Identificación de la Fuente de Adquisición de Imágenes/Vídeos en Escenarios Abiertos usando Técnicas de Inteligencia Artifical

TRABAJO FIN DE MÁSTER
MÁSTER EN INGENIERÍA INFORMÁTICA
CURSO 2017–2018

Ignacio Gago Padreny

Directores

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Calificación: 7,5

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Septiembre de 2018

UNIVERSIDAD COMPLUTENSE MADRID

AUTORIZACIÓN PARA LA DIFUSIÓN DEL TRABAJO FIN DE GRADO Y SU DEPÓSITO EN EL REPOSITORIO INSTITUCIONAL E-PRINTS COMPLUTENSE

Los abajo firmantes, alumno y tutores del Trabajo Fin de Máster (TFM) en el Máster de Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el Trabajo Fin de Máster cuyos datos se detallan a continuación. Así mismo, autorizan a la Universidad Complutense de Madrid a que sea depositado en el repositorio institucional (E-Prints) con el objeto de incrementar la difusión, uso e impacto del TFM en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo: ☐ 6 meses ☒ 12 meses

Título del TFM: Identificación de la Fuente de Adquisición de Imágenes/Videos en Escenarios Abiertos usando Técnicas de Inteligencia Artific

Curso académico: 2017 – 2018

Nombre del Alumno: Ignacio Gago Padreny

Tutores del TFM: Luis Javier García Villalba y Ana Lucila Sandoval Orozco
Departamento de Ingeniería del Software e Inteligencia Artificial

Ignacio Gago Padreny

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Agradecimientos

Quiero expresar mi agradecimiento a mis directores, Luis Javier García Villalba y Ana Lucila Sandoval Orozco, sin su ayuda este trabajo no sería posible. También quiero agradecer el apoyo y ayuda a todo el Grupo GASS .

Este Trabajo Fin de Máster ha sido realizado dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación RAMSES (Internet Forensic Platform for Tracking the Money Flow of Financially-Motivated Malware) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-FCT-2015/700326-RAMSES).

Índice General

Índice de Figuras

Índice de Tablas

Resumen

El creciente uso de nuevas tecnologías en las últimas décadas ha llevado a que los teléfonos móviles sean los dispositivos más utilizados a diario. Como consecuencia, todos los días se toman millones de imágenes y vídeos, generando una gran cantidad de información que puede ser utilizada como evidencia en tribunales. Por tanto, es completamente necesario introducir mecanismos para garantizar la identificación de la fuente de lo que puede ser considerado como prueba en procesos judiciales. En este trabajo se propone un algoritmo para la identificación de la fuente de vídeos, basado en imperfecciones inherentes que los sensores presentan debido al proceso de fabricación, lo que permite distinguir entre dos móviles del mismo modelo y marca. Estas imperfecciones son extraídas de los fotogramas relevantes de los vídeos a través la transformada de ondícula de Daubechies y mediante agrupamiento los vídeos son asociados en distintas clases basados en la correlación. Para determinar el número óptimo de grupos se utiliza el método del codo.

Palabras clave: Agrupamiento Jerárquico, Identificación de la Fuente, Método del Codo, Multimedia Forense, PRNU, Ruido del Sensor, Transformadas de Ondícula, Vídeo Digital.

Abstract

The increasing use of new technologies over the last few decades has come with mobile phones being the most employed device on a daily basis. As a consequence, millions of images and videos are generated every day, originating a huge amount of information that could possibly be used as evidence in court. Therefore, it is completely necessary to introduce procedures that guarantee the identification of the origin from what can be considered proof in forensic matters. In this work an algorithm to identify the source of a video is proposed, based on inherent imperfections that each sensor present due to the manufacturing process, which allows to distinguish between two different mobiles from the same model and brand. These imperfections are extracted from the relevant frames from the videos with Daubechies Wavelet Transform and by means of clustering are grouped into different classes based on their correlation. The elbow method is used to identify the optimal number of clusters.

Keywords: Digital Video, Elbow Method, Hierarchical Clustering, Multimedia Forensics, PRNU, Sensor Noise, Source Identification, Wavelet Transforms.

Lista de Acrónimos

CBR	Tasa de Bits Constante
CCD	Dispositivo de Carga Acoplada
CFA	Matriz Filtro de Color
CMOS	Semiconductor Complementario de Óxido Metálico
DCT	Transformada del Coseno Discreta
FPN	Patrón Fijo de Ruido
G-PRNU	Foto-Reacción Verde No Uniforme
GOP	Grupo de Imágenes
OFG	Gradiente del Flujo Óptico
PCE	Energía Pico de Correlación
PIV	Velocidad de partículas en Imagen
PNU	Pixels No Uniformes
PRG	Gradiente Residual de Predicción

PRNU	Foto-Reacción No Uniforme
SVM	Máquinas de Vector Soporte
VBR	Tasa de Bits Variable

Capítulo 1

Introducción

Desde finales de 1950 con el inicio de la Revolución Digital han sucedido numerosos avances que han colocado la tecnología en un ámbito clave y puntero. Esto ha conllevado una creciente competitividad en el sector dando como resultado mejores productos, con menor coste, y constantes innovaciones en forma de nuevos componentes. A mediados de 2017, se registraron 5.7 mil millones de usuarios únicos en telefonía móvil [1] con una proyección de que para 2020 tres cuartos de la población mundial tendría al menos un teléfono móvil. Paralelamente a este avance, se ha producido un auge en el uso de las redes sociales y de contenido multimedia. Solamente en YouTube, cada año se reproduce contenido equivalente a 46000 años, aproximadamente mil milliones de horas diarias, se sube 400 horas de nuevo contenido cada minuto y el 70 % del tráfico es móvil [2].

El amplio uso de los teléfonos móviles y la mejora y abaratamiento de los sensores fotográficos han situado a las imágenes y vídeos digitales en una de las principales y más grandes fuentes de datos e información, lo que inevitablemente ha supuesto, como no podía ser de otra forma, un aumento en herramientas de edición de imagen y vídeo al alcance de todo el mundo. De esta forma, han surgido numerosas técnicas de falsificación y manipulación de contenido multimedia y con ello un nuevo campo de multimedia forense en continuo estudio y avance.

Al ser utilizados los vídeos como pruebas en procesos judiciales, ya sea cámaras de vigilancia o de dispositivos móviles, es necesario garantizar ciertas cualidades del mismo para poder ser considerado verázmente como evidencia. Una de estas cualidades indispensables es la identificación del origen o autoría del vídeo o imagen, que podría ser comparable a la prueba balística en armas. Como se puede ver, el análisis forense ha ampliado su alcance en los últimos años con la incorporación a la vida cotidiana de imágenes y vídeo, siendo la multimedia un campo de gran relevancia en diversos ámbitos como el judicial. La Sección 1.1 presenta el objeto de la investigación de este trabajo. En la Sección 1.2 se comenta el contexto de la investigación. Finalmente, en la Sección 1.3 se describe la estructura del resto del presente trabajo.

1.1. Objeto de la Investigación

Dentro del análisis forense multimedia, se han realizado numerosas investigaciones centradas en la identificación de la fuente en imágenes, con resultados excelentes. Cabe pensar que al estar un vídeo formado por un conjunto de imágenes, que se presentan de forma que el cerebro las percibe de forma continua, existan también otros tantos trabajos sobre vídeo con buenos resultados. Sin embargo, los altos niveles de compresión que existen al generarse un vídeo conllevan una gran pérdida de información que dificulta la labor de identificación del origen por lo que apenas hay literatura en este respecto.

La identificación de la fuente se puede realizar a partir de unas imperfecciones únicas que cada sensor posee. Estas imperfecciones afectan directamente al proceso de generación de fotogramas y pueden extraerse de estos.

Este trabajo está enfocado en la extracción de fotogramas estratégicos de vídeos y la obtención de la huella o ruido del sensor mediante transformadas de ondícula para identificar la fuente en escenarios abiertos.

A diferencia de los escenarios cerrados, en estos escenarios no se conoce de antemano el conjunto de dispositivos ni se tiene una base de datos con huellas de ciertos dispositivos.

El objetivo que se persigue en este trabajo es, dado un conjunto de vídeos desconocidos, agruparlos en clases según el dispositivo que los ha generado. No entra dentro del alcance identificar la marca y modelo concreto de cada una de estas clases.

1.2. Contexto

El presente Trabajo Fin de Máster se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326)y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (España)

1.3. Estructura del Trabajo

La presente memoria está organizada en 7 capítulos, siendo este el primero.

En los primeros capítulos (Capítulo 2, Capítulo 3 y Capítulo 4) se introducen los conceptos necesarios y el estado del arte para comprender el método propuesto (Capítulo 5), validado por los experimentos realizados (Capítulo 6).

En el Capítulo 2 se introducen los conceptos esenciales sobre los vídeos, como son el proceso de creación de un vídeo basado en el muestreo y la cuantificación, el almacenamiento de los mismos mediante la compresión y la extracción del ruido en fotogramas o imágenes.

En el Capítulo 3 se presenta el análisis forense multimedia y sus distintas técnicas. En concreto se detalla el estado del arte para la extracción de fotogramas claves, para la detección de manipulaciones o de doble compresión y para la identificación de la fuente.

El Capítulo 4 trata sobre algoritmos de agrupamiento. Introduce dos tipos distintos de agrupamientos, como son K-means y agrupamiento jerárquico, distintos métodos para la elección del número óptimo de grupos y métricas de evaluación en agrupamiento basadas en clasificación multiclase.

En el Capítulo 5 se describe el método propuesto, esto es, un algoritmo de extracción de fotogramas con alto grado de información y un algoritmo de agrupamiento basado en la extracción del ruido del sensor mediante la transformada de ondícula de Daubechies que permite identificar la fuente de vídeo.

La experimentación que valida el método propuesto se detalla en el Capítulo 6.

El Capítulo 7 indica las conclusiones extraídas en este trabajo y las futuras líneas de investigación en este ámbito.

Capítulo 2

Vídeos Digitales

En este capítulo se describen los principales conceptos sobre vídeos relacionados con el objetivo principal de este trabajo. En la Sección 2.1 se detalla el proceso de generación de un vídeo y su composición basada en imágenes, para luego hablar de los métodos más habituales de compresión para el almacenamiento del mismo en la Sección 2.2 Una vez explicado este proceso, en la Sección 2.3 se comentará cómo interviene el tipo de sensor en la extracción del ruido o huella digital del dispositivo.

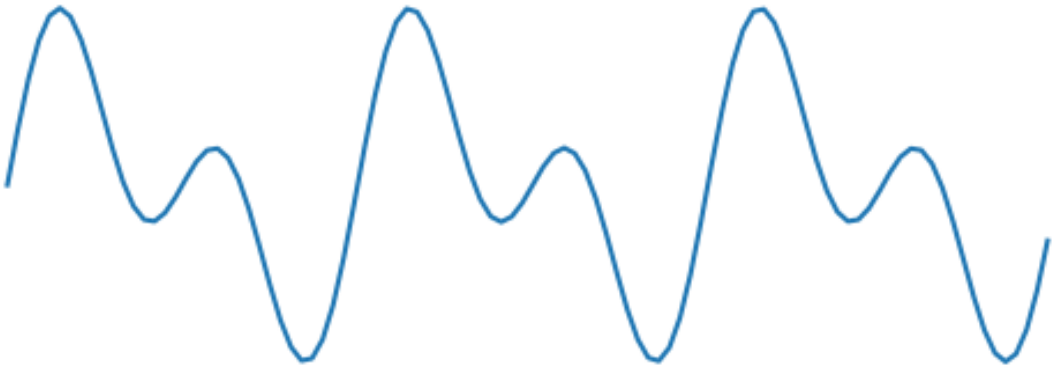
2.1. Proceso de Generación de un Vídeo Digital

El proceso de generación de un vídeo digital está basado en transformar señales analógicas (funciones con dominio continuo y que toman valores en un conjunto continuo) en señales digitales, capaces de ser procesadas por un ordenador. Para convertir una señal analógica en una señal digital (conversión A/D) se utilizan dos técnicas: muestreo y cuantificación.

2.1.1. Técnica de Muestreo

El proceso de muestreo o *sampling* consiste en transformar una señal con dominio continuo en otra de dominio discreto, de forma que se retenga el máximo posible de

la información original de la señal analógica. Gráficamente se puede ver un ejemplo de muestreo en la Figura 2.1, donde se toma el valor de una señal en un número finito de instantes que permite reconstuir adecuadamente la original.



(a) Señal con continuo dominio

(b) Muestreo de la señal

Figura 2.1: Muestro de una señal de dominio continuo

La técnica del muestreo ha sido ampliamente estudiada pues es usada en una gran variedad de campos y existen métodos y fórmulas matemáticas para determinar cotas inferiores que no eliminen información del original. Sin embargo, en este contexto hay cotas menos exigentes debido a la capacidad que tiene el cerebro para procesar visualmente un objeto. En este trabajo hay dos tipos diferentes de muestreo que se deben realizar: uno asociado a las variables espaciales y otro asociado al tiempo. Ambos casos se basan en tener muestras muy cercanas de forma que la composición parezca continua y no discreta. El muestro de la variable temporal está relacionado con el número de imágenes por segundo que es capaz de procesar el ojo humano, estando entre 25 y 30 lo que el ojo ya percibe como continuo.

Al discretizar la señal analógica obtenemos un conjunto finito que podemos

numerar y expresar en forma de una matriz de dos dimensiones, siendo cada una de las celdas un pixel (del inglés *picture element*). Para el número de filas y columnas elegido por el muestro se toma un múltiplo de dos, puesto que tiene por una parte la ventaja de favorecer el direccionamiento de las muestras y por otra de ser más eficientes para ciertos algoritmos como puede ser la transformada de Fourier.

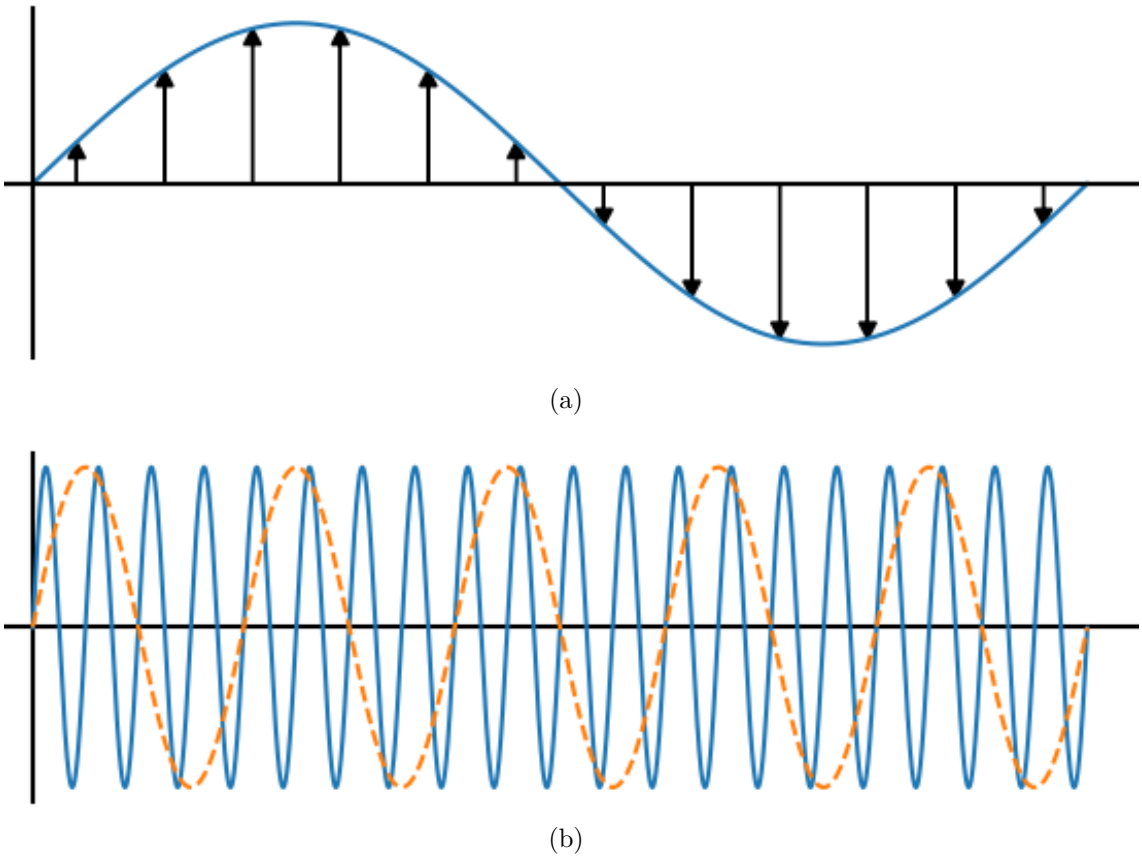
En el muestro también interviene la frecuencia de la señal original: una señal con baja frecuencia puede ser bien representada con una tasa de muestreo determinada, pero la misma tasa de muestreo puede ser no válida para una señal de alta frecuencia, produciéndose el efecto que conocemos como solapamiento o *aliasing*. El teorema de Nyquist establece que utilizando una tasa de muestro mayor al doble de la frecuencia original, se evita el *aliasing* y se puede recuperar la señal original a partir de la transformada.

En la Figura 2.2 se puede observar como cuando la frecuencia de muestreo es suficientemente grande comparado con la frecuencia original (Figura 2.2(a)) se puede reconstruir la onda original, mientras que en la Figura 2.2(b) se observa que cuando no se cumple el Teorema de Nyquist se produce una pérdida de información que impide reconstruir la señal original, obteniéndose a través del muestreo una señal que no representa en absoluto la original [11].

2.1.2. Técnica de Cuantificación

Mientras que el muestreo permite transformar dominios continuos en discretos, la cuantificación consiste en transformar el rango continuo de la señal analógica en un rango discreto. La intensidad captada por el sensor, que es una señal continua, es transformada a un conjunto finito que son los valores que pueden tomar los píxeles. De esta forma, mientras que con el muestro se reduce una variable espacial continua en una matriz, la cuantificación permite que la intensidad que capta la lente del dispositivo se pueda representar por un conjunto discreto de valores.

De la misma forma que en el muestreo, se suele utilizar un conjunto de cardinalidad potencia de dos. Para imágenes en color lo usual es trabajar con tres componentes cada uno de ocho bits, mientras que en las imágenes en blanco y negro se trabaja con un componente de ocho bits. Cabe destacar que este proceso no es



(a)

(b)

Figura 2.2: Frecuencia en el muestreo [11]

reversible y está asociado a funciones no lineales, al contrario que el muestro, en el que partiendo de premisas no muy exigentes se puede reconstruir la señal analógica.

El proceso de cuantificación en vídeo se basa en aplicar fotograma a fotograma el método que se aplica en JPEG.

El primer paso es descomponer cada imagen o fotograma en bloques disjuntos de 8x8 píxels. Cada uno de estos bloques debe expresarse como la suma ponderada de los componentes que se pueden ver en la Figura 2.3. Como se puede observar, los componentes de la esquina superior izquierda representan pocas variaciones y se corresponden con un nivel de detalle pequeño y ondas de baja frecuencia, mientras que los de la esquina inferior derecha tienen un alto nivel de detalle correspondiente a ondas de alta frecuencia.

El peso de cada uno de estos componentes se calcula mediante la Transformada del Coseno Discreta (DCT) siguiendo la ecuación 2.1 [19]:

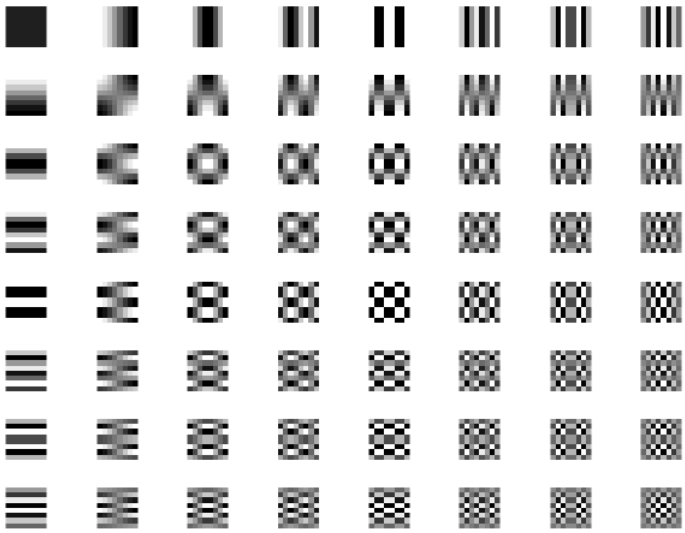


Figura 2.3: Base de cosenos de la DCT

$$D_{ij} = \sum_{k,l=0}^{7} a_{kl}(i,j) B_{kl} \tag{2.1}$$

donde,

$$a_{kl}(i,j) = \frac{1}{4} w(k) w(l) \cos \frac{k(2i+1)\pi}{16} \cos \frac{k(2j+1)\pi}{16} \tag{2.2}$$

y

$$w(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } k = 0 \\ 1 & \text{en caso contrario} \end{cases}$$

Aplicando DCT se transforma la fuente original en el dominio de las frecuencias. Los coeficientes a_{kl} de la ecuación 2.2, los multiplicadores de los valores del bloque de la imagen, cumplen que a medida que se distancian de la primera fila se incrementa la varianza, como se puede ver en la Figura 2.4. Además, a medida que se alejan de la primera columna también crece la varianza. Por otra parte, los coeficientes DCT que se corresponden con frecuencias bajas son grandes en magnitud.

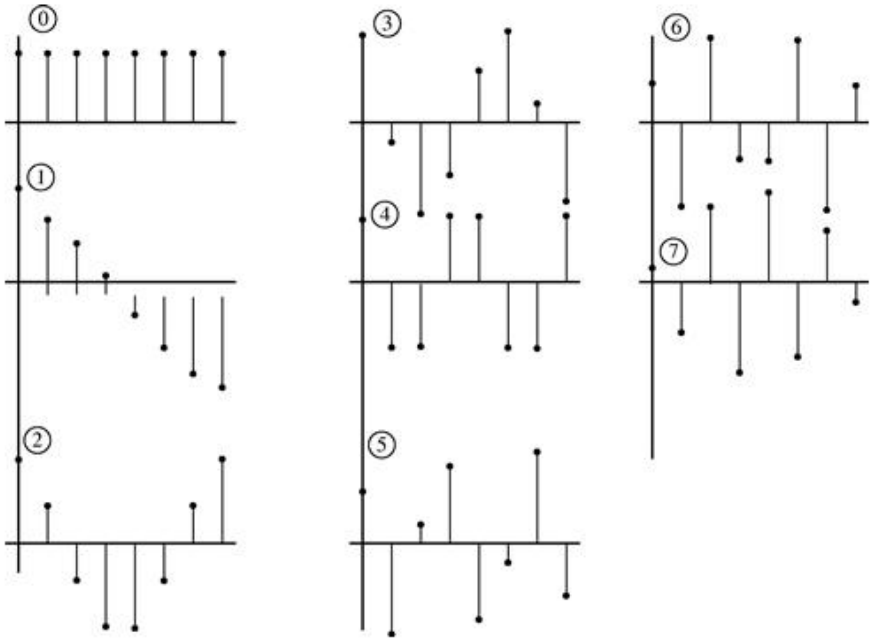


Figura 2.4: Varianza según la fila del bloque en la transformada DCT [23]

La matriz D (ecuación 2.3) con los coeficientes DCT es discretizada posteriormente utilizando una matriz de cuantificación Q. Esta matriz es producto de una tabla de valores de cuantificación y una escala de cuantificación, que es constante en el caso de una Tasa de Bits Variable (VBR) y variable en el caso de una Tasa de Bits Constante (CBR).

$$D_{ij} = round\left(\frac{D_{ij}}{Q_{ij}}\right), i, j \in \{0, \ldots, 7\} \tag{2.3}$$

En la matriz de cuantificación, cada elemento define el umbral bajo el cual un detalle en la imagen debe ser capturado como tal o descartado. De esta forma, a medida que nos alejamos del origen, ya sea horizontal o verticalmente, se exige un mayor coeficiente DCT para que el detalle sea relevante, puesto que se corresponden

con valores de mayor frecuencia. Esto se debe a que el ojo humano tiene mayor dificultad en captar los pequeños detalles y por consiguiente se exige que tengan gran peso en la composición para ser almacenados y no desdechados.

Hay una gran variedad de matrices de cuantificación, calculadas normalmente en base a experimentos psico-visuales para determinar los umbrales DCT. Una matriz de cuantificación utilizada con mucha frecuencia se muestra en la Figura 2.5 [11].

$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

Figura 2.5: Matriz de cuantificación

2.2. Almacenamiento de Vídeos Digitales

Del proceso descrito en la sección anterior, es fácil deducir que la cantidad de datos en señales visuales es grande. Una imagen en blanco y negro de dimensiones MxN con B bits para el nivel de resolución del gris supone un tamaño de $N \times M \times B$ bits. Esto supone que una sola imagen de color de $512 \times 512 \times 8$ ocupa cerca de 1MB. Esto implica que un vídeo con estas características y una tasa de muestro de 30 fotogramas por segundo (el mínimo para que el ojo humano lo detecte como continuo) requiere 23.6MB por segundo [10].

Esta gran cantidad de datos necesaria para almacenar un vídeo no solamente supone un problema en cuanto a requisitos de memoria, si no también para el procesamiento y trasmisión de los mismos. Como consecuencia, es necesario reducir la cantidad de datos mediante algoritmos de compresión, que en el caso de vídeos

están definidos por el comité MPEG (del inglés *Moving Pictures Expert Group*) de forma estándar e internacional.

Como ya se ha comentado anteriormente, se puede ver un vídeo como una sucesión de imágenes o fotogramas. Además de aprovechar la compresión de imágenes, en el caso del vídeo se tiene una redundancia temporal ya que el siguiente fotograma tiene mucho en común con el actual y los anteriores, factor que se aprovechará para reducir el tamaño.

La mayoría de los algoritmos de compresión de vídeo se basan en el concepto llamado Grupo de Imágenes (GOP) (del inglés *Group Of Pictures*). Un GOP de tamaño N está compuesto de N imágenes que pueden ser cuatro diferentes tipos de fotogramas: I, P, B y D.

- **Fotogramas I**: del inglés *intra-coded frames*. Se codifican de forma independiente, sin referencias a otros fotogramas. Esto permite acceso aleatorio a los datos del vídeo, puesto que pueden ser decodificados sin necesitar otros fotogramas. Además de esto, tienen la ventaja de evitar la propagación de errores que se acarrea en la compresión de fotogramas consecutivos al contener la mayor información de la escena por si solos, a costa de ocupar más que los otros tipos de fotogramas. Cada GOP debe tener al menos un fotograma I.
- **Fotogramas P**: son fotogramas pronosticados, comprimidos basados en la diferencia que existe respecto de un fotograma I o fotograma P anterior.
- **Fotogramas B** son fotogramas bidireccionales que usan los datos de imágenes previas y posteriores de fotogramas I o fotogramas P.
- **Fotogramas D**: son fotogramas de baja resolución que raramente se utilizan y que se obtienen decodificando el coeficiente *dc* de la transformada de coseno discreta (DCT) de los coeficientes de cada macrobloque.

En cuanto a la compresión:

- **Fotogramas I**: se comprimen mediante el uso de la transformada del coseno discreta y la cuantificación, de la misma forma que en el caso de imágenes, puesto que estos fotogramas deben contener toda la información relevante de manera aislada. Se comprime por separado la luminosidad y la crominancia.

- **Fotogramas P**: la compresión depende de la similitud entre el fotograma en cuestión y los del grupo en que se encuentra. Si no se encuentra un fotograma adecuado, este deberá comprimirse del mismo modo que si se tratase de un fotograma I. En caso de encontrarse un buen candidato, se calcula el residuo entre ambos y se cuantifica utilizando la matriz de la Figura 2.6.

$$\begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix}$$

Figura 2.6: Matriz de cuantificación de fotogramas tipo P

- **Fotogramas B**: utilizan el mismo procedimiento que los fotogramas P con la diferencia que también buscan similitud con fotogramas posteriores en el grupo, y también pueden utilizar la relación entre un fotograma anterior y uno posterior simultáneamente.

Una vez se tiene la cuantificación del fotograma, independientemente del tipo que sea, éste se almacena siguiendo una traza en forma de zig-zag (ver Figura 2.7), y no de forma secuencial, agrupando los ceros correspondientes a los coeficientes de alta frecuencia en un mismo grupo [9]. Posteriormente, la codificación se realiza mediante el algoritmo de Huffman [25].

El procesamiento de un GOP no es secuencial, al existir relaciones bidireccionales entre cierto tipo de fotogramas. Al empezar un GOP, en primer lugar se procesa el fotograma tipo I. El siguiente fotograma a procesar será de tipo P, puesto que solamente necesita de este fotograma tipo I. Una vez procesados estos dos fotogramas, los fotogramas tipo B que están en medio serán decodificados. El proceso sigue alternando el procesamiento de fotogramas tipo P con fotogramas tipo B intermedios, hasta finalizar el GOP en cuestión, como se puede ver en la Figura

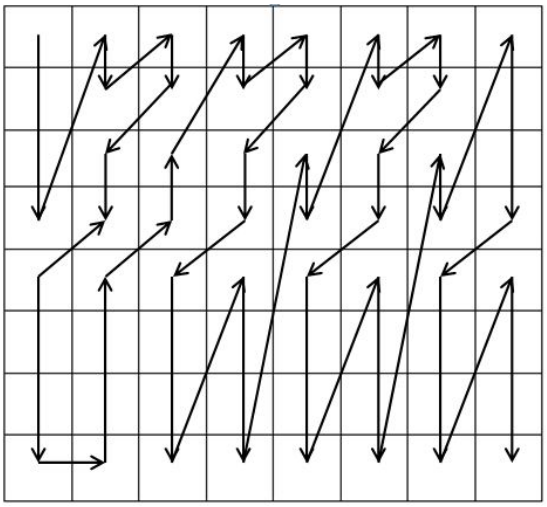


Figura 2.7: Método del zig-zag

2.8.

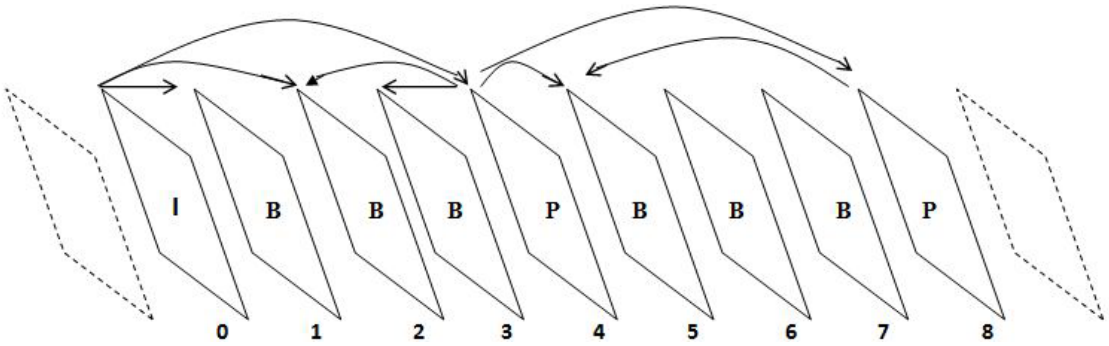


Figura 2.8: Procesamiento en Group of Pictures

2.3. Procesamiento de Imágenes en los Sensores

Existen principalmente dos tipos de sensores que se usan para capturar imágenes o vídeo: sensores de tipo Dispositivo de Carga Acoplada (CCD) (del inglés *Charge Coupled Device*) y sensores de tipo Semiconductor Complementario de Óxido Metálico (CMOS) (del inglés *Complementary Metal Oxide Semiconductor*). Ambos sensores se basan en el mismo principio, capturar la máxima cantidad de luz que incide en el sensor y convertirla en una señal eléctrica que será transformada posteriormente en digital. Los sensores CMOS tratan los píxeles de forma individual

mientras que los sensores CCD se basan en la propagación de carga eléctrica mediante condensadores.

En la actualidad, los sensores CMOS son ampliamente utilizados, sobre todo en dispositivos móviles, ya que los sensores CCD necesitan un chip adicional y son más costosos y grandes que los CMOS. A continuación, se detalla el funcionamiento de los sensores CMOS [24].

2.3.1. Sensores CMOS

Un sensor CMOS está formado por una matriz de sensores de píxeles, cada uno de ellos compuesto por un fotodetector y un amplificador activo. Cada uno de estos sensores de píxeles captura información de un píxel en uno de los tres colores primarios (rojo, verde y azul) puesto que se aplica un filtro de color conocido como Matriz Filtro de Color (CFA) (del inglés *Color Filter Array*).

El filtro de color más utilizado es el filtro de Bayer. Está compuesto por un patrón de filtro que es la mitad verde, un cuarto azul y un cuarto rojo, debido a que el ojo humano es más sensible al color verde [11], en la Figura 2.9 un ejemplo de filtro de Bayer.

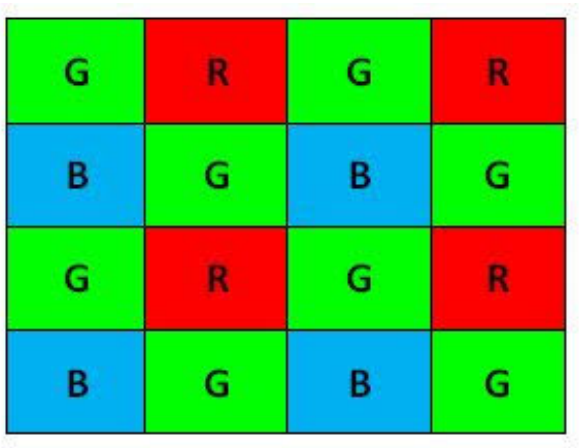


Figura 2.9: Filtro de Bayer

Tras la aplicación del filtro CFA para cada píxel se tiene únicamente información sobre un color, lo que implica que se tiene que llevar a cabo un proceso para estimar los valores de los otros dos componentes del color. Esta estimación se puede realizar mediante distintas técnicas, todas ellas basadas en utilizar los valores de los píxels cercanos. Se pueden usar métodos sencillos como el del vecino más próximo (el píxel toma el valor del píxel que le precede) o el bilinear (toma como valor la media de sus vecinos en vertical y horizontal más próximos) u otros más complejos como pueden ser *splines* cúbicas, método de mínimos cuadrados o filtros direccionales [10].

Tras la conversión Bayer-RGB hay varias funciones que en función del dispositivo y sensor pueden aplicarse como pueden ser corrección del color o corrección gamma,

entre otros. Hay que tener en cuenta que en este proceso el hardware tiene una influencia considerable. Cada sensor a pesar de pertenecer al mismo fabricante tiene pequeñas imperfecciones o diferencias del resto que impactan directamente en la imagen obtenida. Esto es conocido como el ruido del sensor, que se aborda en la siguiente sección.

2.4. Extracción del Ruido en Imágenes

Los principales componentes del ruido en imágenes por imperfecciones del sensor son el Patrón Fijo de Ruido (FPN) (*Fixed Pattern Noise*) y el Foto-Reacción No Uniforme (PRNU) (*Photo Response Non Uniformity*).

El ruido FPN se genera por la corriente oscura y depende también de la exposición y de la temperatura. Es un ruido independiente de las imperfecciones del sensor y es un ruido aditivo que es eliminado en algunas cámaras restando una capa de color negro.

El ruido PRNU es el mayoritario y es un ruido multiplicativo, lo que complica su eliminación. Está compuesto por dos ruidos: Pixels No Uniformes (PNU) (*Pixel Non-Uniformity*) y por defectos de baja frecuencia como pueden ser la conFiguración del zoom, y la refracción de la luz en las lentes. Es el primero de estos dos componentes, el PNU, el que tiene que ver con la fabricación de los wafers de silicio y las imperfecciones en el proceso de fabricación, lo que hace que sea un atributo único de cada sensor. La extracción del PRNU se basa en aplicar una función a la imagen original que elimine el ruido de esta, obteniendo como resultado la imagen limpia de ruido. Al substraer la imagen original de la misma sin ruido obtenemos el ruido. En [5] usan el algoritmo BM3D y en [6] proponen usar el algoritmo FSTV, en [3] usan transformadas de ondículas o *wavelets* para eliminar el ruido. Este ruido puede estar contaminado por agentes externos, en consecuencia se han desarrollado técnicas como *zero-mean* [8] o el filtro de Wiener [12].

Capítulo 3

Análisis Forense en Vídeos Digitales

En este capítulo se presentan las principales técnicas forenses aplicadas a vídeos. A pesar de ser un campo ampliamente investigado, la gran cantidad de vídeo que se produce a diario y el crecimiento de aplicaciones de edición de vídeo para usuarios no expertos ha crecido exponencialmente en los últimos años. Esto hace que muchos de los algoritmos desarrollados queden desactualizados frente a nuevas técnicas antiforenses, que buscan no ser detectados por técnicas forenses ya existentes.

La gran digitalización de la sociedad en las últimas décadas ha influido de forma notable en procesos judiciales, especialmente posteriormente a 1978 puesto que tras la legislación de Florida se admitían pruebas digitales (e-mails, fotografías o vídeos digitales, audios, etc.) como evidencia. Para garantizar que estas pruebas digitales puedan considerarse como evidencia de sucesos reales, tiene que garantizarse que no hayan sido manipuladas y debe poder establecerse la fuente de adquisición de los datos digitales en cuestión.

En las siguientes secciones se describen las principales áreas de investigación del análisis forense en vídeos: la detección de manipulación de vídeos, la detección de doble compresión (un caso concreto que permite la detección de manipulación de vídeos) y la identificación de la fuente. Sin embargo, primero es necesario definir cómo se puede medir el grado de similitud entre dos imágenes, dado que en muchas

de las técnicas forenses es necesario comparar fotogramas o regiones de fotogramas.

3.1. Similitud entre Fotogramas

Detectar cuándo dos fotogramas son iguales o muy parecidos es una parte importante del análisis forense en imágenes y vídeos. Además de ser una herramienta muy útil en la detección de inserciones de fotogramas u objetos en vídeos, también permite asegurar los derechos de autor y obtener los fotogramas más representativos de un vídeo. Con los fotogramas menos similares, se puede resumir un vídeo a través de la indexación y la navegación. Los métodos más utilizados para analizar la similitud entre fotogramas son los histogramas de color y *perceptual hashing*.

3.1.1. Histogramas de Color

Este método se basa en que las imágenes o fotogramas se corresponden con una serie de valores de píxel, e imágenes similares tendrán proporciones parecidas de ciertos colores [50]. Al agrupar en clases y representar las frecuencias, el histograma no se verá afectado al manipular la orientación, tamaño o posición de la imagen.

Algunas técnicas simples obtienen un 90 % de precisión para detectar imágenes similares en bases de datos [50]. Para cada canal RGB se crea un histograma y se utiliza la distancia euclídea para comparar cada clase (en inglés *bin*).

Técnicas más elaboradas utilizan un único histograma en lugar de tres, pero cada clase del histograma tiene información que relaciona todos los canales [51]. Además, también realizan pruebas con otras variables distintas al color. Para calcular la similitud entre los histogramas, se utilizan diferentes métodos como la distancia euclídea o la intersección de histogramas. La frecuencia fue normalizada en base al número de píxeles de cada imagen para abarcar el redimensionamiento de imágenes.

La principal desventaja que tiene el uso del histograma de color es que no tienen en cuenta características espaciales o geométricas: el mismo objeto cambiado de color no será visto como el mismo. Además, cambios en la cuantificación pueden pasar

desapercibidos por algoritmos basados en el histograma de color.

3.1.2. Perceptual Hashing

El uso de funciones *hash* o funciones resumen ha sido ampliamente utilizado en criptografía. Estas funciones se han caracterizado por transformar dos entradas muy similares en salidas completamente distintas. Sin embargo, las funciones de tipo *perceptual hashing* obtienen resultados muy similares para imágenes de entrada muy parecidas, de forma que la distancia entre la salida de dos *perceptual hash* puede ser utilizado para comparar dos imágenes. La distancia que se utiliza es la distancia de Hamming, que cuenta el número de bits que difieren.

Los algoritmos para calcular el *perceptual hash* de una imagen utilizan pasos similares. A continuación, se describen los más comunes: *average hash*, *difference hash* y *phash* [53].

- ***Average hash*** sigue los siguientes pasos:
  1. **Reducir el tamaño**: se reduce la imagen a una de tamaño 8x8. De esta forma se reducen elementos de alta frecuencia y los detalles de la imagen.
  2. **Reducir el color**: se convierte a escala de grises.
  3. **Calcular el color medio**.
  4. **Calcular el hash**: se compara cada píxel con el color medio.
- ***Difference hash***: sigue los mismos dos primeros pasos que *average hash* pero en vez de comparar el valor del píxel con el color medio lo compara con el píxel situado a su derecha. Genera menos falsos positivos que *average hash*.
- ***Phash***: el más complejo y más costoso computacionalmente. Consta de los siguientes pasos:
  1. **Reducir el tamaño**: se reduce la imagen a una de tamaño 32x32.
  2. **Obtener la luminosidad**: para cada píxel resultante se obtiene el valor de la luminosidad.
  3. **Transformada DCT**: se aplica la transformada DCT a la matriz resultante del paso anterior.

4. **Extraer las bajas frecuencias**: de la matriz de 32x32 obtenida en el paso 3, se eliminan las altas frecuencias y se toma únicamente los 8x8 valores correspondientes con frecuencias bajas, los de más arriba y más a la izquierda.
5. **Calcular el hash**: se compara cada píxel con la mediana.

En [52] se comparan distintos métodos, entre ellos algunos basados en histogramas de color, frente a su propuesta que consisten en dos fases: en la primera extraen fotogramas en base a la entropía y en la segunda utilizan *phash*, obteniendo buenos resultados.

3.2. Manipulación de Vídeos

Existen multitud de herramientas de edición de vídeo a disposición de usuarios no expertos, que les permiten de una forma muy sencilla manipular un vídeo: agregar, quitar, clonar o copiar fotogramas y aplicar otras operaciones como rotar, escalar o aplicar filtros. En cualquiera de los casos, a pesar de que no se haya añadido o eliminado ningún objeto en los fotogramas, la codificación del vídeo se verá afectada.

La aplicación de técnicas forenses de imágenes para la detección de manipulaciones en vídeos no es recomendable. La estructura de los vídeos y los distintos tipos de fotogramas del GOP permiten usar técnicas basadas en la consistencia temporal incapaces de ser detectadas por medio de análisis estáticos de los fotogramas considerados como imágenes. Además, son algoritmos computacionalmente costosos incapaces de detectar la inserción o eliminación de fotogramas [14].

En [26] analizan el problema de la duplicación de fotogramas y utilizan la correlación como medida de similitud. En primer lugar, se eligen un conjunto de segmentos candidatos de entre todos los del vídeo, reduciendo el espacio de búsqueda. Posteriormente, se calcula la medida de similitud entre ellos a partir de histogramas de color. Finalmente se decide si se ha producido duplicación o no. Para los experimentos se utilizaron un conjunto de 15 vídeos manipulados, y el algoritmo tuvo una precisión media del 85 %.

En [31] se centran en la detección de eliminación o duplicación de fotogramas consecutivos. Para ello se basan en la Velocidad de partículas en Imagen (PIV) (del inglés *Particle Image Velocity*), su objetivo es comparar fotogramas adyacentes y estimar el desplazamiento causado por la separación en el tiempo. La duplicación o eliminación de fotogramas consecutivos aumenta este desplazamiento. Para decidir si un desplazamiento concreto se debe a una manipulación, utilizan el test de Grubbs [32] que para una distribución normal permite detectar *outliers*. En los experimentos realizados con parten de 40 vídeos a partir de los cuales generan otros 40 eliminando fotogramas y otros 40 duplicando fotogramas. La precisión media es del $96,3\,\%$ con una tasa de falsos positivos del $10\,\%$.

En [33] se demuestra que la correlación entre los coeficientes de valores grises es consistente en vídeos pero cuando se produce una falsificación esta consistencia desaparece. Primero, se extrae la consistencia de la correlación de los coeficientes de los valores grises para posteriormente clasificar las características usando Máquinas de Vector Soporte (SVM) (del inglés *Support Vector Machines*). En los experimentos parten de una base de datos de vídeos originales y crean otras cuatro bases de datos a partir de los originales con las siguientes manipulaciones: insertando 25 fotogramas, insertando 100 fotogramas, eliminando 25 fotogramas y eliminando 100 fotogramas. Para entrenar la SVM se utilizan 480 vídeos originales y 480 vídeos falsificados, dejando 118 originales y otros tantos manipulados para pruebas, consiguiendo precisiones por encima del $90\,\%$.

Los mismos autores, en [34] utilizan otro método basado también en la consistencia temporal entre fotogramas. En este caso en lugar de utilizar la correlación entre los coeficientes grises, utilizan la consistencia medida mediante el flujo óptico de Lucas-Kanade que permite determinar el movimiento de un objeto dentro de una secuencia de fotogramas. Para los experimentos se utilizan la mismas bases de datos que en el caso anterior y una SVM para la clasificación, consiguiendo también una precisión media superior al $90\,\%$.

En [42] se basan en el flujo óptico pero lo calculan siguiendo Horn-Schunck en lugar de Lucas-Kanade. Se extraen únicamente los fotogramas tipo I y tipo P y extraen como características el Gradiente Residual de Predicción (PRG) (del inglés *Prediction Residual Gradient*) y el Gradiente del Flujo Óptico (OFG) (del inglés *Optical Flow Gradient*). El PRG se centra en variaciones en la posición de

objetos mientras que el OFG está centrado en cambios en la luminosidad. Estas dos características son comparadas con unos umbrales determinados empíricamente para detectar picos, cuando los picos sean continuos se tratará de una manipulación. El método ha mostrado una precisión de un 86 % en las pruebas realizadas.

En [41] se utiliza también el flujo óptico de Lucas-Kanade para detectar inconsistencias en el caso de manipulaciones de tipo inserción o eliminación de fotogramas. En lugar de calcular la correlación entre fotogramas del flujo óptico primero utilizan un estadístico que resume la información de los vectores resultantes de Lucas-Kanade para comprobar la consistencia con estos. Cuando se detectan irregularidades en base a este estadístico, se calcula la correlación entre los vectores completos del flujo Lucas-Kanade. En los experimentos se utilizan un total de 115 vídeos con una precisión del 90 %.

Algunos autores han desarrollado métodos basados en la extracción de algún tipo de huella a partir de los fotogramas que componen el vídeo para detectar anomalías en fotogramas con huellas significativamente distintas. Estos métodos tienen el incoveniente de que los vídeos comprimidos pierden mucha información sobre la huella, y solamente han demostrado dar buenos resultados en vídeos no comprimidos que suele ser poco usual.

En [15] obtienen el PRNU de los primeros fotogramas que componen el vídeo y se utilizan distintas medidas para detectar ataques como inserción de fotogramas, inserción de objetos y clonación de fotogramas mediante la correlación entre el PRNU de referencia del vídeo y el ruido de un fotograma en concreto, la relación entre el ruido de dos fotogramas consecutivos o la relación entre dos fotogramas consecutivos.

En [16] utilizan en lugar del PRNU un ruido que solamente es aplicable a los sensores CCD, el ruido del fotón en disparo. Además, el método solamente es aplicable a vídeos grabados de forma estática sin la cámara en movimiento lo cual restrige mucho el ámbito de aplicabilidad del algoritmo. En el caso en el que se den las premisas de las que parten el método tiene una precisión del 97 %.

Aprovechando la consistencia entre fotogramas de tipo temporal, algunas investigaciones se han centrado en aspectos de tipo geométrico como pueden ser que las propiedades físicas o de iluminación sean reales.

En [17] se utilizan técnicas geométricas para detectar trayectorias imposibles de objetos en vuelo libre. Para ello, se modeliza el movimiento parabólico de estos objetos en tres dimensiones para proyectar este modelo posteriormente en dos dimensiones y compararlo con la trayectoria de ese mismo objeto en el vídeo. El método desarrollado es válido tanto para cámaras estáticas como para cámaras en movimiento. Los experimentos se han realizado con diversos vídeos ya sea generados por ellos u obtenidos de plataformas de compartición de contenido en los que han medido el error medio entre la trayectoria real y la trayectoria estimada mediante su procedimiento para ser capaces de clasificar cuando una trayectoria ha sido falseada. Sin embargo, no hay datos sobre la precisión del algoritmo en cuestión. Hay que tener en cuenta que esta técnica solamente puede ser utilizada en vídeos en los que exista un objeto que describa una trayectoria parabólica y por tanto no es válida para cualquier vídeo.

Sin embargo, la mayoría de trabajos sobre la detección de manipulación de vídeos están basados en detectar la re-comprensión o doble compresión de un vídeo, puesto que al ser editado se vuelve a comprimir por segunda vez.

3.3. Doble Compresión

Gran parte de los estudios sobre doble compresión se centran en vídeos con formato MPEG y utilizan las mismas ideas que en la detección de doble compresión de imágenes JPEG. En concreto, la re-cuantificación afecta a los coeficientes DCT al usar un paso de cuantificación distinto del original, víendose en el histograma de los coeficientes DCT [14]. Estos coeficientes pueden ser aproximados como se indica en la ecuación 3.1 [18]

$$Y_{Q_1,Q_2} = \Delta_2 sign(Y) round\left(\frac{\Delta_1}{\Delta_2} round\left(\frac{|Y|}{\Delta_1}\right)\right) \tag{3.1}$$

donde Δ_1 y Δ_2 son el tamaño del paso en la primera y segunda compresión, respectivamente.

En [19] se demuestra como la relación que hay entre Δ_1 y Δ_2 influye en

el histograma creando un máximo característico. Intuitivamente, la idea es que al descomprimirse la imagen, modificarse una porción de la misma y volverse a comprimir, esa porción modificada mostrará trazas de una sola compresión mientras que el resto de la imagen tendrá rasgos de doble compresión.

En [27] se presenta un método para el caso en el que se ha utilizado la misma matriz de cuantificación en ambas compresiones, basado en el número de coeficientes DCT distintos que hay en una compresión, doble compresión y triple compresión.

En [30] utilizan un conjunto de clasificadores binarios entrenados con distintas combinaciones entre Δ_2 que es conocida (se puede leer directamente de los datos del vídeo) y posibles Δ_1, para luego tomar el voto de la mayoría con las características concretas del vídeo en cuestión.

En [38] se afirma que se obtiene el mismo resultado comprimiendo la imagen una vez con Δ_1 que si se comprime dos veces con la misma matriz de cuantificación Δ_1. De esta forma, dada la imagen original se comprime nuevamente con distintas matrices hasta encontrar una que cumpla que en la mayoría de los bloques codificados se obtenga la imagen original, obteniendo así la matriz de cuantificación que se utilizó en la última compresión. La manipulación se detecta cuando existen algunos bloques distintos entre la imagen comprimida y la original, puesto que entonces la última compresión no ha sido aplicada por igual en todos los bloques y ha existido una doble compresión. Para la experimentación se utilizó un conjunto de 1338 imágenes y dentro de las pruebas que realizaron la peor precisión media que se obtuvo fue del $88,7\,\%$. Es importante tener en cuenta que si la imagen ha sido manipulada y se ha utilizado la misma matriz de cuantificación que en la primera compresión, este método no logrará detectarlo.

En [39] se calculan las diferencias entre los coeficients DCT extráidos en cuatro direcciones: horizontal, vertical, diagonal mayor y diagonal menor. Tras obtener estas cuatro matrices se truncan algunos elementos basados en ciertos umbrales para luego modelarse por medio de un proceso aleatorio de Markov de primer orden. Tras algunas transformaciones sobre estos procesos de Markov, se crea un vector de características que será procesado por algoritmos de *machine learning* puesto que la doble compresión conlleva unos errores de redondeo que dejan muestras estadísticas caracterizables por Markov. Para la experimentación se generaron 5040 vídeos con

la misma estructura GOP y con distintas combinaciones Δ_1 y Δ_2, obteniendo una precisión superior al 90 %.

Otra estrategia ampliamente utilizada se basa en que mientras que los coeficientes DCT de una imagen siguen una distribución Laplace [36] o una distribución Cauchy [37], también siguen la ley de Benford [35]. Esto es, el primer dígito significativo es d con probabilidad $\log_{10}\left(1+\frac{1}{d}\right)$. Si los coeficientes DCT se alejan significativamente de esta distribución entonces se puede concluir que se ha utilizado doble compresión. Muchas investigaciones han utilizado procedimientos basados en la ley de Benford aplicados al caso de doble compresión JPEG en imágenes, extensibles a vídeos.

En [21] se parte de la hipótesis que el factor de multiplicación q_1 de la tabla de cuantificación y el factor q_2 de la segunda tabla de cuantificación varían mientras que la tabla se mantiene la misma. En el caso de que $q_1 > q_2$ se observan anomalías en el histograma de los coeficientes DCT de tipo AC (frecuencia distinta de cero en ambas dimensiones espaciales) distintos de cero y es posible detectar la doble compresión.

En [20] se aplica la ley de Benford para un subconjunto de las frecuencias del DCT que identifican más sensible al número de compresiones y utilizan un multiclasificador compuesto de N clasificadores SVM S_k con $(k = 1, \ldots, N)$ donde S_k es un clasificador binario que detecta si la imagen ha sido comprimida k veces o no. De esta forma el número de compresiones que ha sufrido la imagen se toma como el mayor k tal que el clasificador S_k ha detectado que ha sido comprimido k veces. Para la experimentación se utilizó un conjunto de prueba de 100 imágenes y un conjunto de test de 10 imágenes, obteniendo una precisión del 94 % considerando que como máximo podía haber $N = 4$ compresiones.

En [40] se aplica la ley de Benford para vídeos puesto que la codificación MPEG para vídeos y JPEG para imágenes comparten el mismo proceso. Se observa que cuando se utiliza VBR los coeficientes AC distintos de cero de los fotogramas tipo I doblemente comprimidos se alejan de la ley de Benford solamente cuando la escala de cuantificación utilizada en Δ_2 es menor que la utilizada en Δ_1. En el caso de CBR se aprecian compartamientos anómalos sin distinción de casuística. Para formalizar lo anterior, se utilizan clasificadores binarios SVM en los que tratan como unidad un GOP decidiendo que existe doble compresión si

el porcentaje de fotogramas detectados como doblemente comprimidos excede un umbral determinado, obteniendo una precisión media superior al 95 %.

Los métodos descritos arriba se basan en extender técnicas desarrolladas para imágenes para el caso de vídeo. Sin embargo, existen otros algoritmos que aprovechan la alteración de la estructura GOP de los vídeos. Dentro de un GOP, los fotogramas tipo P están correlacionados con el fotograma tipo I inicial, de forma que en caso de existir doble compresión los fotogramas que cambien de GOP, ver Figura 3.1 mostrarán ciertas características estadísticas.

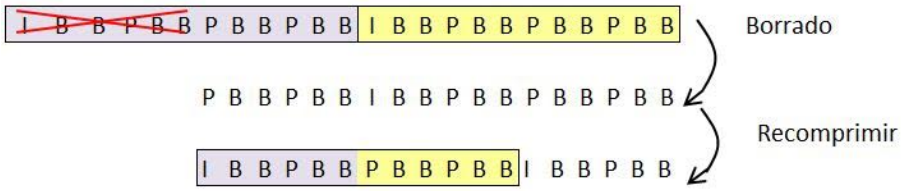


Figura 3.1: Reestructuración de GOP tras doble compresión, [14]

En [22] se analiza el error de movimiento o compensación de movimiento de los fotogramas tipo P, esto es, la transformación que se debe aplicar a un fotograma de tipo I o de tipo P anterior para obtener el fotograma tipo P en cuestión. Al deshacerse la secuencia original GOP, existirán fotogramas tipo P cuyo fotograma de referencia haya cambiado y el error de movimiento en ese caso es mayor. En este trabajo no se describe el método para calcular el umbral que determine si el error de movimiento corresponde con un cambio en la estructura GOP ni tampoco se dan resultados concretos sobre la precisión del algoritmo.

En [28] se analiza las características periódicas de la cadena de bits de datos y del *skip macroblocks* para todos los fotogramas tipo I y tipo P. Los *skip macroblocks* son usados en fotogramas tipo P y tipo B y no contienen información, y se corresponden con macrobloques en los que no se producen cambios respecto del fotograma tipo I sobre el que se codifican. Al cambiar la estructura GOP, el número de *skip macroblocks* decrece puesto que el fotograma I original en el que se basaba el fotograma P era antes un fotograma tipo P.

En [29] se utiliza el número de *inter-coded macroblocks* y de *skip macroblocks* de cada fotograma modelizados como $i(n)$ y $s(n)$. Cuando se produce un pico en $s(n)$ hay una alta probabilidad de doble compresión y el fotograma I del que toma

los valores fuera anteriormente un fotograma de tipo P, utilizando un razonamiento análogo al caso anterior.

3.4. Identificación de la Fuente

La identificación de la fuente de adquisición es de vital importancia para muchos procesos judiciales, podría compararse con las pruebas balísticas para identificar un arma. Es por esto por lo que la identificación de la fuente en imágenes ha sido ampliamente estudiado por académicos en los últimos años con buenos resultados. Esta sección se restringe a la identificación de la fuente entendido como la identificación del modelo fuente en dispositivos móviles y no engloba otras temáticas como podría ser distinguir entre gráficos generados por ordenador o capturados.

Existen pocas investigaciones en vídeos a pesar de que un vídeo se descompone como una secuencia de fotogramas. Sin embargo, la menor resolución en vídeo frente a imagen y las altas compresiones que se utilizan hacen que se pierda mucha información sobre la huella.

En [44] se extraen una serie de fotogramas del vídeo en base a la luminosidad para extraer el PRNU mediante la descomposición *wavelet* de Daubechies de cuarto nivel a los que se aplica el filtro de Wiener. Se calcula la correlación entre el ruido de cada fotograma para posteriormente evaluarlo mediante la Energía Pico de Correlación (PCE) (del inglés *Peak-to-Correlation Energy*). Se utiliza un método de clasificación en el que los fotogramas a analizar son caracterizadas en uno u otro grupo según la PCE.

En [45] tratan el vídeo como una secuencia de N fotogramas, para cada uno de los fotogramas extraen el PRNU y se utiliza el estimador de máxima verosimilitud para identificar el PRNU del vídeo. Para decidir si dos vídeos fueron tomados por la misma cámara se basan en la covarianza normalizada y en la PCE. Si provienen del mismo dispositivo entonces la PCE es grande por el pico en la covarianza normalizada y en caso de no provenir de la misma fuente la covarianza normalizada parecerá ruido blanco. En la experimentación se utilizaron 25 cámaras y se muestra como el nivel de compresión del vídeo es crucial para el algoritmo, entre mayor compresión menor

calidad y más tiempo de vídeo (en algunos casos 10 minutos de vídeo) se necesita para obtener un PRNU suficientemente bueno, lo que hace que este método no sea efectivo para vídeos de corta duración grabados por móvil.

En [46] se utiliza un subconjunto de los coeficientes AC de la transformada DCT formado a partir de tres índices p, q y r que toman 8 orientaciones diferentes. Para cada una de esas orientaciones se calculan 9 estadísticos en base a la relación de orden entre p y q y entre r y q lo que da un total de 72 estadísticos diferentes que denominan caracerísticas CP, también utilizados en otros trabajos de estegoanálisis, que utilizarán como *input* para un clasificador de tipo SVM. En los experimentos utilizan 4 modelos de cámara diferentes y 10 vídeos de cada una de ellas, obteniendo una precisión del 100 %.

En [43] se propone el uso de características propias de la codificación MPEG-2, características relacionadas con la tasa de bits, los factores de cuantificación y los vectores de movimiento. Tanto la tasa de bits como los factores de cuantificación y los vectores de movimiento no son parámetros fijos en el estándar MPEG-2, cada fabricante establece unos en concreto según el sensor. Tras extraer estas características, se utiliza un clasificador SVM entrenado. Para las pruebas utilizan vídeos de ocho codificadores distintos y obtienen precsiones por encima del 86 %. Hay que tener en cuenta que vídeos obtenidos de cámaras que compartan el mismo codificador de MPEG-2 no serán clasificados como distintos por lo que este método solamente sirve para garantizar que dos vídeos provienen de distinta fuente.

En [7] se propone utilizar el canal verde por ser el que tiene más información sobre la huella. El algoritmo propuesto extrae el canal verde de la imagen y mediante interpolación bilineal se redimensionan los fotogramas a tamaño 512x512. Posteriormente, se extrae el ruido mediante *soft-thresholding*. El PRNU del vídeo se obtiene como la media de los ruidos de cada fotograma y se clasifican utilizando la correlación como medida de similitud. En los experimentos se muestra cómo los resultados de este proceso con la Foto-Reacción Verde No Uniforme (G-PRNU) (*Green PRNU*) son mejores que con el PRNU.

Capítulo 4

Técnicas de Agrupamiento

Agrupamiento (*clustering*) es una técnica de aprendizaje no supervisado que a partir de un conjunto de datos y una medida de similitud o distancia entre ellos se agrupan en distintos grupos o clases de forma que elementos que están en el mismo grupo son más parecidos entre ellos que respecto a los de otro grupo distinto.

En este capítulo se describen las principales técnicas de agrupamiento y los métodos de elección del número óptimo de grupos.

4.1. Técnica de K-Means

El algoritmo de agrupamiento K-means agrupa n elementos en k clases S_k con el objetivo de minimizar la suma del error cuadrático intra-grupo de la ecuación 4.1.

$$\sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{4.1}$$

donde μ_i es la media de las distancias entre los elementos en S_i, también conocidos como centroides.

La suma del error cuadrático intra-grupo o inercia es un indicador de la cohesión de los grupos. A mayor cohesión, menor distancia existe entre los elementos de cada clase y por tanto también del centroide. Este indicador, sin embargo, tiene ciertas desventajas:

- Asume que los grupos se modelan como esferas, al estar basado en k centroides y minimizar la distancia euclídea, lo que implica misma varianza entre grupos (ver la Figura 4.1).
- No es una métrica normalizada
- Muy sensible a outliers al utilizar la distancia al cuadrado
- No tiene en cuenta la densidad de cada grupo. Implícitamente asume que al ocupar cada clúster el mismo área cada grupo tiene que tener el mismo número de puntos (ver Figura 4.2).

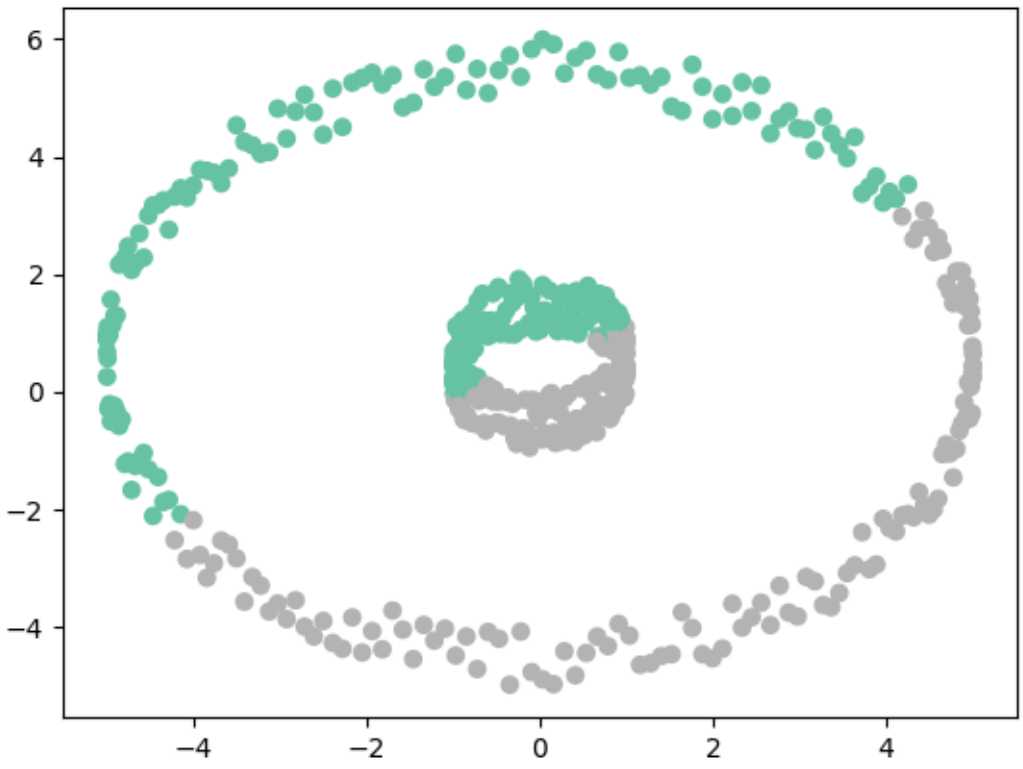


Figura 4.1: K-means frente a varianza en grupos

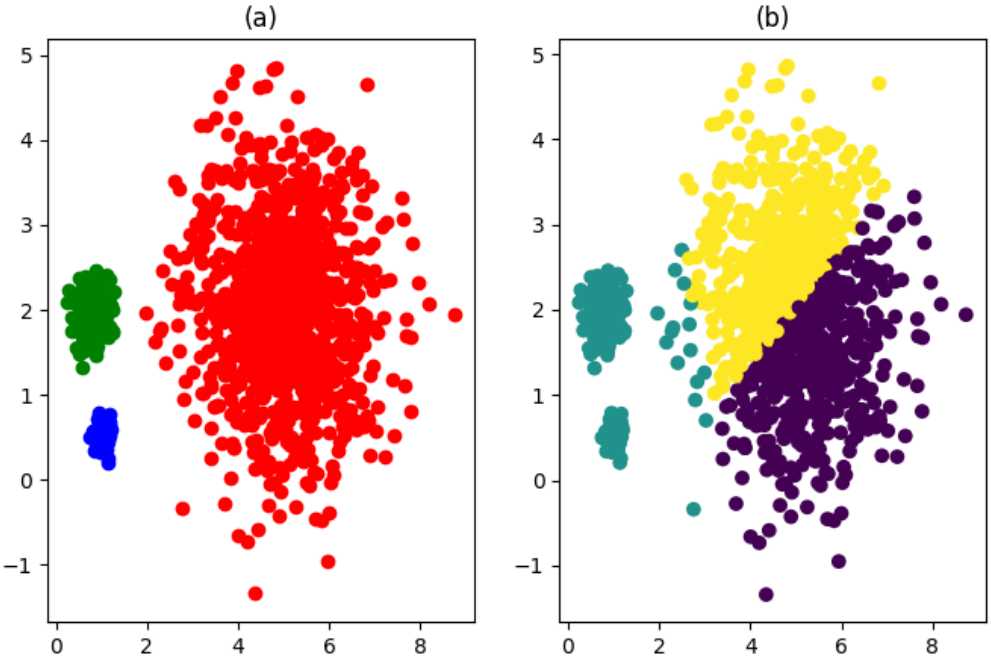


Figura 4.2: Agrupamiento real, Agrupamiento obtenido por K-means

Dado un conjunto de n elementos $\{x_1, x_2, \ldots, x_n\}$ K-means empieza con una fase de inicialización en la que se escogen k centroides, $\{c_1, c_2, \ldots, c_k\}$. Una vez elegidos los centroides, itera de la siguiente forma [54]:

- **Etapa de asignación**: asigna cada elemento al centroide que minimiza la distancia euclídea al cuadrado, el grupo S_i que tiene como centroide c_i de la ecuación 4.2

$$S_i = \{x_j : \|x_j - c_i\|^2 \leq \|x_j - c_p\|^2 \; \forall p, 1 \leq p \leq k\} \tag{4.2}$$

- **Actualización de los centroides**: media de los elementos del grupo correspondiente (ver ecuación 4.3).

$$c_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j \tag{4.3}$$

Cuando en la etapa de asignación no se producen cambios entre dos iteraciones consecutivas, el algoritmo termina.

La inicialización de los centroides es un aspecto muy relevante en el algoritmo: dado que K-means converge cuando encuentra un óptimo local se han desarrollado diversos métodos de inicialización para encontrar el óptimo global:

- **Aleatorio**: se asigna de forma aleatoria un elemento a cada grupo y posteriormente se calculan los centroides en base a esta asignación. Este método ubica los centroides cerca del centro del conjunto de datos.
- **Forgy**: se escogen al azar k elementos del conjunto y se utilizan como centroides. Este método tiende a dispersar los centroides iniciales [56].
- **MacQueen**: escoge al azar k elementos del conjunto y los trata como centroides. Asigna cada elemento al grupo con el centroide más próximo y recalcula los centroides, que serán los centroides de inicialización para el algoritmo [54].
- **K-means++**: propuesto en el año 2007 [55]. Funciona de la siguiente forma:

  1. Se elige un centroide elegido aleatoriamente sobre el conjunto de observaciones.
  2. Se calcula la distancia al cuadrado entre cada observación y el centroide seleccionado.
  3. Se elige otro punto al azar como segundo centroide, la probabilidad que tiene cada observación de ser elegido es proporcional a la distancia al cuadrado calculada en (2).
  4. Repetir (2) y (3) hasta tener k centroides.

En los últimos años se ha utilizado ampliamente la inicialización mediante K-means++, además de ejecutar varias veces el algoritmo con distintos centroides para evitar óptimos locales.

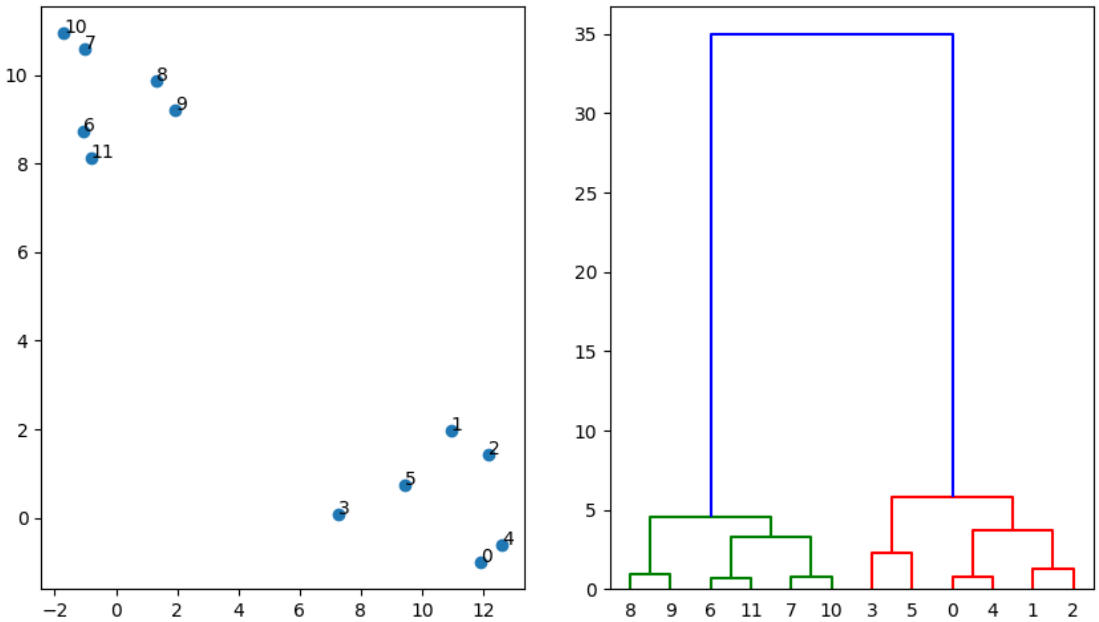
4.2. Agrupamiento Jerárquico

El agrupamiento jerárquico se refiere a una familia de algoritmos de agrupamiento que construyen clases de forma anidada al fusionarlos (agrupamientos aglomerativos) o dividirlos (agrupamientos divisivos) [57].

El agrupamiento aglomerativo, construirá un único grupo dado un conjunto de n elementos en n pasos. En cada iteración se fusionarán dos grupos de forma que se minimize la medida de distancia o similitud especificada. El algoritmo termina cuando hay 1 grupo.

En el caso de agrupamiento divisivo, se comienza con una única clase que contiene las n observaciones. En cada una de las n iteraciones se crea un nuevo grupo, hasta tener n grupos.

En cualquiera de los dos casos, las sucesivas iteraciones son representadas a través de un dendrograma. Como se puede ver en la Figura 4.3, un dendrograma es un diagrama en forma de árbol. En la figura se muestra un algoritmo de agrupamiento aglomerativo en el que las observaciones de la Figura 4.3(a) se han ido agrupando sucesivamente hasta terminar en un solo grupo. Se puede ver como las distancias en el dendrograma de la Figura 4.3(b) aumentan a medida que se van formando grupos y se produce un salto importante al juntar los dos *clusters* reales.



(a) Observaciones (b) Dendrograma

Figura 4.3: Dendrograma

En lo sucesivo se contemplará el agrupamiento aglomerativo, los conceptos y métodos para el agrupamiento divisivo son análogos. Para decidir cómo agrupar las clases de forma iterativa es necesario definir una medida de similitud entre grupos, de forma que grupos similares se agrupen antes que grupos distintos. En la primera iteración todos los grupos están compuestos por un único elemento, se especifica una distancia. En el resto de iteraciones es necesario definir un criterio de enlace o *linkage criteria* que modele la medida de similitud entre dos grupos en los que al menos uno de ellos está compuesto por más de una observación. Se puede especificar cualquier distancia d, mientras cumpla las propiedades de distancia desde un punto de vista matemático, que son:

- $d(x, y) \geq 0$ y $d(x, y) = 0 \iff x = y$
- $d(x, y) = d(y, x)$ o propiedad simétrica
- $d(x, z) \leq d(x, y) + d(y, z)$ o desigualdad triangular

Las distancias más habituales son las siguientes [58]:

- **Distancia euclídea**: la más común y la que se utiliza por defecto.

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2} \tag{4.4}$$

- **Distancia del taxi**: conocida como distancia Manhattan debido al diseño en cuadrícula de las calles de la isla.

$$d(x, y) = \sum |x_i - y_i| \tag{4.5}$$

- **Distancia de Chebyshev**: conocida como distancia del tablero de ajedrez ya que coincide con el número de movimientos que necesita el rey para moverse de una casilla a otra.

$$d(x, y) = \text{máx} |x_i - y_i| \tag{4.6}$$

- **Distancia de Hamming**: para vectores lógicos, es el número de bits que tienen que cambiarse para transformar un vector de bits en otro.

$$d(x, y) = \frac{c_{01} + c_{10}}{n} \tag{4.7}$$

donde c_{ij} es el número de ocurrencias de $x[k] = i, y[k] = j$ para $k < n$.

- **Distancia de Mahalanobis**: distancia muy útil para determinar la similitud entre dos variables aleatorias multidimensionales al tener en cuenta la correlación y la escala de ellas.

$$d(x, y) = \sqrt{(x - y)V^{-1}(x - y)^T} \tag{4.8}$$

donde V es la covarianza y V^{-1} la inversa de la matriz de covarianza.

Además, otras distancias ampliamnete utilizadas son Bray-Curtis, Canberra, coseno, Minkowski, o la euclídea normalizada. Para el caso de variables booleanas, además de la ya mencionada arriba distancia Hamming se han empleado: dado, Jaccard-Needham, Kulsinski, Rogers-Tanimoto, Russell-Rao, Sokal-Michener, Sokal-Sneath y Yule.

Una vez definidas la distancia a utilizar entre cada par de observaciones, se pueden definir distintos criterios de enlace entre dos clústers u y v. Los más habituales son los siguientes:

- **Método *single***: método del punto más cercano. Este algoritmo se centra en la separación entre grupos pero no en la cohesión de los mismos y permite formas geométricas más flexibles que en otros casos. Sigue la ecuación 4.9

$$d(u, v) = \text{mín}\left(dist(u[i], v[j])\right) \tag{4.9}$$

para todos los puntos i en el grupo u y todos los j en v.

- **Método *complete***: también conocido como Algoritmo del punto lejano o Algoritmo de Voor Hees. La distancia se calcula mediante la ecuación 4.10

$$d(u, v) = \text{máx}\left(dist(u[i], v[j])\right) \tag{4.10}$$

- **Método *average***: o también algoritmo UPGMA. Cumple la ecuación 4.11

$$d(u, v) = \sum_{ij} \frac{dist(u[i], v[j])}{(|u| * |v|)} \tag{4.11}$$

- **Método *weighted***: se conoce también como el algoritmo WPGMA. Se calcula

a partir de la ecuación 4.12

$$d(u,v) = (dist(s,v) + dist(t,v))/2 \tag{4.12}$$

donde el grupo u está compuesto por los *clusters* s y t.

- **Método *centroid***: asigna como distancia entre grupos la distancia euclídea entre sus centroides. Sigue la ecuación 4.13

$$d(u,v) = d(\bar{x}_u, \bar{x}_v) \tag{4.13}$$

donde $\bar{x}_u = \frac{1}{n}\sum_i u[i]$.

- **Método *ward***: según este método la distancia entre dos grupos u y v es el incremento que se producirá en la suma del error cuadrático si se fusionan. En este caso, a diferencia de K-means, el número de puntos interviene en la fórmula por lo que dados dos pares de grupos cuyos centros están distanciados por igual, el algoritmo de Ward fusionará los de menor cardinalidad. Se define mediante la ecuación 4.14

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T}d(v,s)^2 + \frac{|v|+|t|}{T}d(v,t)^2 - \frac{|v|}{T}d(s,t)^2} \tag{4.14}$$

donde u es el nuevo grupo creado a partir de s y t y $T = |v| + |s| + |t|$.

4.3. Técnicas de elección del Número de Grupos Óptimo

Los algoritmos de agrupamiento se utilizan principalmente con dos propósitos distintos según la problemática:

- Se conoce a priori el número de distintas clases en la población y se quieren obtener los cortes en el vector de características de las observaciones para conocer qué características tiene cada clase. Nuevas observaciones podrán ser categorizadas a partir del conocimiento extraído en el agrupamiento.
- No se conoce cuántas clases distintas existen en la población y se quiere

conocer esto a partir del agrupamiento. En este caso en el que no se tiene información sobre el k a utilizar en el algoritmo K-means o el corte a aplicar en el dendrograma en agrupamiento jerárquico. A continuación se describen distintos métodos basados en heurísticas para determinar el número óptimo de grupos.

4.3.1. Coeficiente Silueta

Para cada observación x se tienen dos medidas:

- **Cohesión**: grado de similitud del elemento x respecto del grupo. Se obtiene como la distancia promedio de x a todos los puntos en el mismo grupo.
- **Separación**: grado de disimilitud del elemento x respecto a elementos que han sido identificados en otras clases. La separación más utilizada es la distancia promedio entre x y todos los elementos del grupo más cercano, aunque también se utilizan otras medidas para valorar la separación.

El coeficiente silueta para x se define con la ecuación 4.15

$$s(x) = \frac{b(x) - a(x)}{\text{máx}\{a(x), b(x)\}} \tag{4.15}$$

donde $a(x)$ es la cohesión y $b(x)$ la separación. Para todo el agrupamiento está definido por la ecuación 4.16

$$SC = \frac{1}{n} \sum_{x} s(x) \tag{4.16}$$

donde n es el número de observaciones.

Intuitivamente, un agrupamiento bien definido debería corresponderse con que para cada elemento x se tiene que $a(x) << b(x)$, es decir, x está muy cercano respecto de los elementos de su grupo en comparación con los elementos del grupo más cercano. El coeficiente $s(x)$ toma los valores en el rango $[-1, 1]$, donde -1 corresponde con una mala elección del número de grupos y 1 indica grupos bien definidos. De esta forma, una de las técnicas que se usa con el coeficiente silueta es elegir un rango de valores para k, y elegir k de forma que el coeficiente silueta para el agrupamiento sea máximo.

4.3.2. Índice Calinski-Harabasz

Dado k, se define el índice de Calinski-Harabasz con la ecuación 4.17:

$$s(k) = \frac{SS_B}{SS_W} * \frac{N-k}{k-1} \tag{4.17}$$

donde SS_B es la varianza entre grupos y está definida por la ecuación 4.18

$$SS_B = \sum_{i=1}^{k} n_i d(m_i, m)^2 \tag{4.18}$$

donde m_i es el centroide del grupo i y m es la media de todas las observaciones. SS_W es la varianza intra-grupo (ver ecuación 4.19):

$$SS_W = \sum_{i=1}^{k} \sum_{x \in S_i} d(x, m_i)^2 \tag{4.19}$$

Para que los grupos estén bien definidos deben tener valores grandes para SS_B (medida de separación) y pequeños para SS_W (medida de cohesión). De esta forma el índice de Calinski-Harabasz es una adaptación del método F-test de ANOVA, SS_B con $k-1$ y SS_W con $n-k$ grados de libertad por lo que aparecen en la fórmula pues SS_B debe ser proporcional a $k-1$ y SS_W proporcional a $n-k$. El método a seguir es el mismo que en el caso del coeficiente silueta, elegir un rango de valores para k y elegir el k que maximice el coeficiente Calinski-Harabasz. Al igual que el coeficiente silueta, es un método que funciona generalmente bien para grupos convexos.

4.3.3. Método del codo

El método del codo consiste en representar en una gráfica el porcentaje de varianza explicada frente al valor k del número de grupos. De forma visual se identifica el valor de k como aquel que si se añadiese otro grupo no mejora o incrementa apenas la varianza explicada. Cabe mencionar que el método del codo no está ligado únicamente a la varianza explicada y pueden utilizarse otras medidas como por ejemplo en agrupamiento jerárquico la distancia mínima entre grupos en cada iteración. En ese caso las distancias serán pequeñas hasta que en un punto aumenten considerablemente al fusionar grupos muy distintos. Este último punto

será el k óptimo buscado. Este método recibe el nombre por la gráfica característica que se produce al representar una de estas variables frente al número de grupos, como puede verse en la Figura 4.4, donde el k óptimo aparece marcado en rojo.

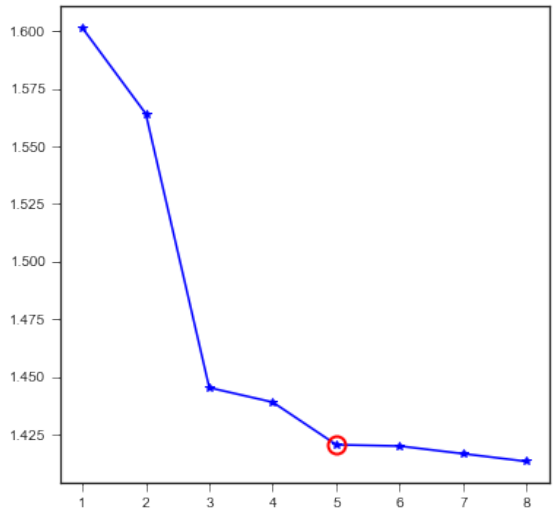


Figura 4.4: Método del codo

4.3.4. Método Gap

El método Gap fue desarrollado por unos investigadores en Stanford [48]. Este método formaliza matemáticamente el método del codo para encontrar el k óptimo. Dado

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{4.20}$$

donde D_k es la suma de la distancia intra-grupo entre sus elementos. En caso de utilizarse la distancia euclídea, W_k es la suma de los cuadrados de las distancias intra-grupo.

La idea es estandarizar el gráfico $\log(W_k)$ mediante la comparación de este con su esperanza bajo una distribución nula de referencia. La ecuación 4.21 define el estadístico Gap.

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k) \tag{4.21}$$

El valor óptimo de k es el mínimo k tal que $Gap(k) \geq Gap(k+1) - s_{k+1}$, donde s es la desviación mínima. En la Figura 4.5 se muestra la similitud entre el método del codo y el de Gap.

4.4. Evaluación del Agrupamiento

En esta sección se presentan algunos métodos utilizados para evaluar la calidad de la agrupación resultante del algoritmo de *clustering*, en concreto se describe la precisión, la exhaustividad y el valor-f, para lo que es imprescindible relacionar los algoritmos de agrupamiento con los de clasificación.

A pesar de existir algunas métricas especialmente diseñadas para algoritmos de agrupamiento, se basan en que no se conoce el número real de grupos. En este caso, puede utilizarse esta información para evaluar el agrupamiento expresando este como si de un problema de clasificación se tratase. Es importante destacar que esta información solamente se utiliza para la evaluación del agrupamiento y no para el algoritmo de agrupamiento.

Un algoritmo de clasificación multiclase, de n clases, trata de clasificar observaciones en una de las n clases. De esta forma, se puede ver un algoritmo de agrupamiento como un algoritmo de clasificación multiclase, puesto que agrupa las observaciones en grupos o clases similares. Como consecuencia, en caso de conocerse a priori el número de grupos original y la clase de cada elemento se pueden utilizar las mismas métricas de evaluación que para los algoritmos de clasificación.

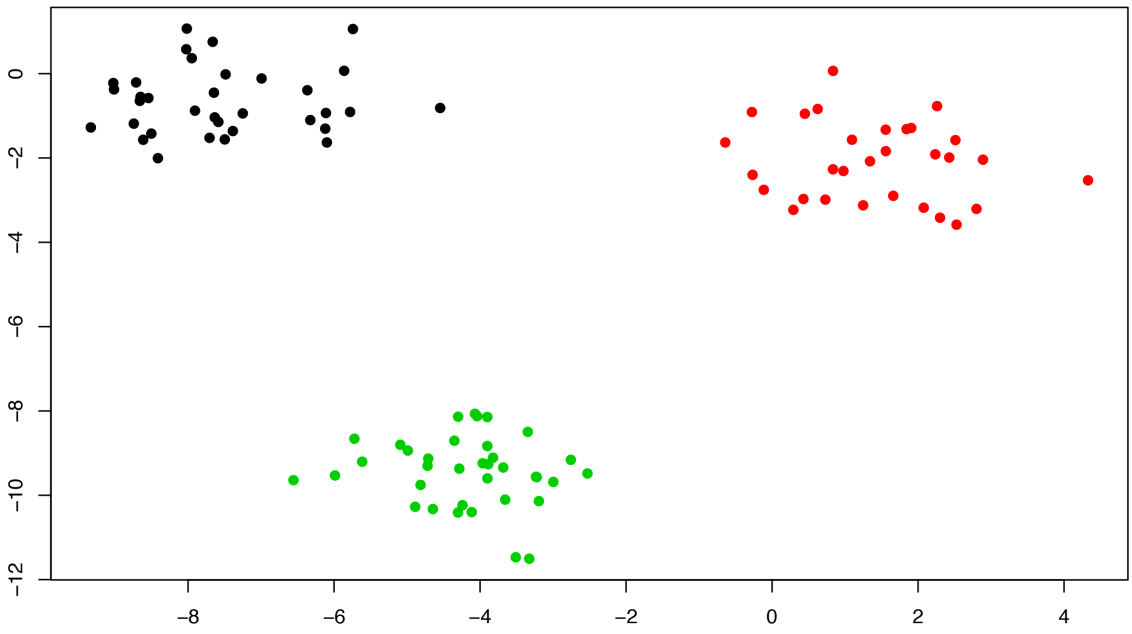
La precisión es el ratio $\frac{TP}{TP+FP}$ donde TP son los verdaderos positivos y FP los falsos positivos. Se puede definir intuitivamente como la habilidad de un clasificador de no clasificar una observación en una clase distinta a la verdadera. La exhaustividad se expresa como el cociente $\frac{TP}{TP+FN}$ donde FN es el número de falsos negativos y es la capacidad del clasificador de encontrar todas las muestras verdaderas. Tanto la precisión como la exhaustividad toman valores en el intervalo $[0, 1]$, donde el valor 1 es el óptimo. El valor-f tiene en cuenta tanto la precisión como la exhaustividad, y al igual que estos toma valores en el intervalo $[0, 1]$ y su mejor valor es el 1, y matemáticamente se define como

$$F = 2 * (p * r)/(p + r)$$

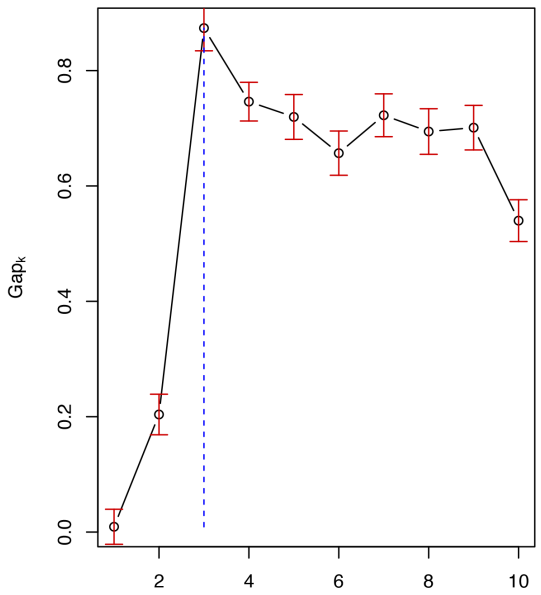
donde p es la precisión y r la exhaustividad.

Estas definiciones están pensadas para el caso de clasificadores binarios, por lo

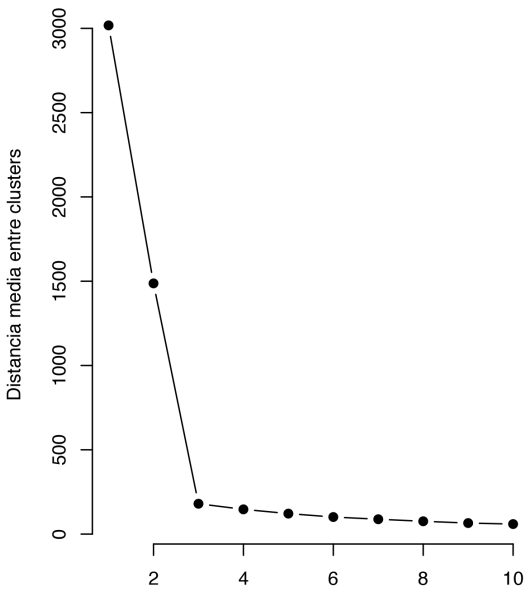
que en el caso de clasificación multiclase se deben calcular de forma individual para cada clase y promediarlos.



(a) Conjunto de observaciones



(b) Método Gap



(c) Método del Codo

Figura 4.5: Método del codo vs método Gap

Capítulo 5

Método propuesto

En este capítulo se presenta un método de identificación de la fuente en vídeos generados por dispositivos móviles en escenarios abiertos, esto es, no se conoce el conjunto de entrada ni se tiene una base de datos de referencia con la que comparar. A partir de un conjunto de entrada de vídeos se obtendrán grupos en función del origen de los mismos.

El procedimiento puede dividirse en varias etapas, en las que se ha tomado la opción que mejores resultados ha proporcionado a partir de la experimentación.

5.1. Consideraciones Generales

La idea principal que subyace es extender los algoritmos existentes que se aplican en imágenes para obtener el PRNU y aplicarlos en vídeos, expresando el vídeo como una sucesión de fotogramas.

El primer paso consiste en extraer los fotogramas del vídeo y sus características. Se extraen los fotogramas de tipo I ya que tienen los menores niveles de compresión y por tanto mayor información. Desde un punto de vista teórico, el uso de la totalidad de los fotogramas tipo I como representación de la huella del vídeo tiene el riesgo de obtener un PRNU sesgado si una o varias escenas predominan en el vídeo ya

que estas escenas influirán en el PRNU de numerosos fotogramas, afectando al promedio que es utilizado como el ruido del vídeo. En este trabajo se propone la extracción de las escenas más disimilares utilizando el *phash* y la distancia de Hamming como similitud. Experimentalmente, en el siguiente capítulo, se analizan ambas metodologías y se observa que los *key-frames* no permiten conseguir una huella fiable del vídeo por lo que se opta por la utilización de la totalidad de los fotogramas de tipo I.

Una vez se han obtenido los fotogramas relevantes (ya sean todos los fotogramas tipo I o los *key-frames*) se extrae el PRNU de cada fotograma mediante el algoritmo desarrollado en [49] basado en la transformada *wavelet* de Daubechies y se define el sensor de la cámara como la media del ruido extraído de los fotogramas clave.

El último paso es aplicar agrupamiento jerárquico utilizando el método Ward puesto que se ha visto en las pruebas que es el que mejor resultado otorga en este contexto. De la misma forma, se ha validado experimentalmente que el método del codo es el que determina con mayor precisión el número de grupos óptimo.

5.2. Especificación del Método

A continuación se describen los pasos que componen el método propuesto. En la descripción se incluye la fase de extracción de fotogramas clave como parte del algoritmo a pesar de que experimentalmente los resultados obtenidos con los fotogramas tipo I son mejores, a modo de conocer los dos distintos algoritmos que se compararán en el siguiente capítulo.

1. Toma como entrada un conjunto de n vídeos correspondientes a m cámaras de móvil distintas.

2. **Extracción de fotogramas de tipo I**: se decodifica cada vídeo y se extraen las características de cada fotograma, en particular si un fotograma es de tipo I, B o P, lo que permite seleccionar los fotogramas de tipo I, que al haber sido comprimidos sin referenciar otros fotogramas contienen mayor grado de información que los fotogramas tipo P o fotogramas tipo B. Como se muestra

mediante experimentos, el alto nivel de compresión de los fotogramas de tipo P y de tipo B evitan la obtención de una huella significativa a partir de estos por lo que es necesario buscar los fotogramas con menor nivel de compresión en vídeo que son los de tipo I. Aún así, la compresión de los fotogramas I es bastante alta en comparación a imágenes.

3. **Extracción de fotogramas clave**: para no contaminar el ruido con la escena, se deben elegir fotogramas que representen las distintas escenas del vídeo. Para ello en primer lugar se calcula el *phash* de cada fotograma y se construye la matriz de distancias $D(i, j)$ para cada par de fotogramas, utilizando la distancia Hamming. Para seleccionar los más disimilares se define un umbral determinado y el primer fotograma tipo I como fotograma clave. Se elijen los demás fotogramas tipo I de forma que la distancia Hamming entre todos ellos sea mayor al umbral.

4. **Extracción del PRNU**: se extrae el PRNU de cada fotograma clave y se calcula el PRNU del vídeo como la media de los PRNU de los fotogramas clave.

5. **Matriz de distancias para agrupamiento jerárquico**: se computa la correlación entre ruidos y se utiliza como matriz de distancias en un algoritmo de agrupamiento jerárquico aglomerativo. Al tratarse de un escenario abierto, no se tiene una base de datos que asocia sensores a ruido PRNU por lo que es necesario agrupar los vídeos de entrada en grupos de forma que cada grupo represente un dispostivo distinto.

6. **Elección del número óptimo de grupos**: el número óptimo de grupos se obtiene mediante el método del codo.

A continuación se detalla el algoritmo de extracción de fotogramas clave.

El primer fotograma al contener la primera escena siempre formará parte del conjunto de fotogramas clave. Para seleccionar el próximo fotograma clave primero se genera una lista de candidatos con los fotogramas cuya distancia al primer fotograma es mayor a la mediana. El fotograma de distancia máxima se añade al conjunto de fotogramas clave y se repite el proceso de generación de candidatos y elección del máximo con la particularidad de que el conjunto de candidatos de la iteracción $i+1$ se interseca con el obtenido en la intersección i, para evitar seleccionar fotogramas que

Algorithm 1 Extracción de fotogramas clave

Input video, secuencia de fotogramas
Output fotogramas clave
$ifotogramas \leftarrow extraerIfotogramas(video)$
$hashes \leftarrow phash(ifotogramas)$
$dist \leftarrow matrizDistancias(hashes)$
$umbral \leftarrow mediana(dist)$
$fotogramas\_clave \leftarrow [0]$
$ult \leftarrow 0$
$f\_candidatos \leftarrow \{(dist(ult, j), j) \in range(1, N) \mid dist(ult, j) > umbral\}$
while $f\_candidatos \neq \emptyset$ **do**
    $max\_dist, ult \leftarrow$ máx$(f_candidatos)$
    $ifotogramas\_clave \leftarrow ifotogramas\_clave \cup \{ult\}$
    $f\_candidatos \leftarrow f\_candidatos \setminus (max\_dist, ult)$
    $f\_candidatos \leftarrow \{(dist(ult, j), j) \in f\_candidatos \mid dist(ult, j) > umbral\}$
return $fotogramas\_clave$

han sido descartados en iteracciones anteriores puesto que no cumplían la condición de la mediana.

El umbral se ha determinado experimentalmente en base a pruebas. El *phash* se construye a partir de 64 píxels por lo que se obtiene un *hash* de 64 bits, lo que implica que la distancia Hamming máxima entre dos fotogramas es 64. Se ha buscado un umbral que dependa del vídeo y se ha optado por la mediana. De esta forma, como poco el 50 % de los fotogramas quedarían descartados.

Los fotogramas extraídos mediante el algoritmo 1 representan las distintas escenas de un vídeo. Tras extraer el PRNU de cada uno de estos fotogramas se define el PRNU como la media del ruido de estos, lo que constituye la entrada para el algoritmo de agrupamiento jerárquico.

El algoritmo de agrupamiento jerárquico utilizará como distancia la matriz de correlaciones con una transformación. La correlación entre el PRNU p_1 y p_2 se define como:

$$\rho_{p_1,p_2} = \frac{\mathrm{Cov}(p_1, p_2)}{\sigma_{p_1}\sigma_{p_2}} \tag{5.1}$$

Se transforma la matriz de correlaciones ρ en una matriz de distancias mediante $1 - \rho$ puesto que:

- $1 - \rho$ es semipositiva.
- En ρ elementos similares toman valores cercanos a 1, lo que implica que la distancia en ese caso tiene que ser cercana a 0.
- En ρ valores muy distintos toman valores cercanos a -1, que son convertidos valores próximos a 2, distancia máxima en $1 - \rho$.
- Las otras dos propiedades de la distancia (además de semipositividad) se satisfacen.

Para la elección del número óptimo de grupos se utiliza el método del codo puesto que otros indicadores como el coeficiente silueta o el índice de Calinski-Harabasz, que como se puede ver en los experimentos dan peores resultados en este contexto. El inconveniente del método del codo es que requiere de la inspección visual de la gráfica y por consiguiente no puede automatizarse.

Capítulo 6

Experimentos y Resultados

En este capítulo se describen las pruebas realizadas en relación a la identificación de la fuente en vídeos. En la Sección 6.1 se realiza un experimento que pretende poner de manifiesto el impacto de la compresión en la calidad de la huella extraída en vídeos, puesto que el método propuesto se basa en la utilización de fotogramas y existen numerosos trabajos con buenos resultados sobre la extracción de la huella en imágenes pero mucha menos literatura en cuanto a vídeo. En la Sección 6.2 se describen las pruebas realizadas que resultaron en la elección de todos los fotogramas tipo I en lugar de solamente los *key-frames*, y en la elección del método del codo sobre otros métodos que se comentan, finalizando esta sección con un ejemplo completo del algoritmo propuesto.

Para los experimentos se utilizan dos conjuntos de datos distintos: uno de imágenes y otro de vídeos. El dataset de imágenes es utilizado en el experimento de la compresión y se puede ver en la Tabla 6.1. El dataset utilizado para los experimentos relacionados con el método propuesto está compuesto por vídeos que han sido generados por dispositivos móviles, y para cada una de las pruebas que se han realizado se escoge al azar el número de dispositivos o móviles a utilizar, tomando para cada uno de ellos a su vez un número aleatorio de vídeos y elegidos también aleatoriamente. Las condiciones en las que se han grabado estos vídeos son las siguientes:

- Máxima resolución del dispositivo.

- Orientación vertical
- Sin zoom
- Vídeos no estáticos
- Dentro y fuera de recintos
- Aproximadamente unos diez segundos
- Condiciones cotidianas

El dataset del que se parte para vídeo se puede ver en la Tabla 6.2.

Cámara	Formato
Canon 60D	.TIF
Nikon D90	.TIF
Nikon D7000	.TIF
Sony A57	.TIF

Tabla 6.1: Cámaras de fotografía

Marca	Modelo	Formato	Tamaño GOP	Vídeos disponibles
BQ	Aquaris E5	.mp4	30	10
Samsung	G6	.mp4	30	10
Iphone	7	.mov	30	10
Huawei	Y635 L01	.mp4	30	10
Iphone	8 Plus	.mov	30	8
Nexus	5	.mp4	31	10
Xiomi	M3	.mp4	30	13

Tabla 6.2: Cámaras de vídeo de móviles

6.1. Experimento Compresión

La identificación de la fuente mediante el ruido del sensor PRNU en imágenes y los buenos resultados que se han presentado en los estudios realizados hace pensar que adaptando esta técnica sobre el conjunto de fotogramas de un vídeo lleve también a excelentes resultados. Este experimento trata de ilustrar el efecto que tiene la compresión en la calidad de la huella que se puede extraer sobre un fotograma o imagen para entender la dificultad que existe en el caso de la identificación de la fuente en vídeos por sus altos grados de compresión.

Puesto que un vídeo está compuesto por múltiples fotogramas o imágenes, este experimento va a utilizar como dataset un conjunto de imágenes de buena calidad con el que se van a generar otros tres datasets, cada uno con un grado de compresión distinto. De esta forma, estas imágenes se pueden ver como los fotogramas de un vídeo que tienen una compresión mucho mayor al original.

En concreto, se parte de imágenes en buena calidad en formato .TIF y se transforman a calidad 90, calidad 85 y calidad 80 formato JPEG. En todos los casos se extrae el PRNU de cada una de las imágenes y se aplica agrupamiento jerárquico utilizando el coeficiente silueta para la elección del número óptimo de grupos.

Sin modificar la calidad de las imágenes el resultado del agrupamiento se puede ver en la Tabla 6.3.

Marca y Modelo	**1**	**2**	**3**	**4**
Canon 60D	**35**	0	0	0
Nikon D90	0	1	**34**	0
Nikon D7000	0	**35**	0	0
Sony A57	0	0	0	**35**

Tabla 6.3: Agrupamiento con imágenes formato .TIF

En la Tabla 6.4 están los resultados para las mismas imágenes en formato JPEG calidad 90 %.

Marca y Modelo	**1**	**2**	**3**	**4**	**5**
Canon 60D	**27**	4	3	1	0
Nikon D90	4	4	**26**	1	0
Nikon D7000	5	4	4	2	**20**
Sony A57	0	**27**	4	4	0

Tabla 6.4: Agrupamiento JPEG calidad 90 %

Como se puede observar la compresión al 90 % empeora bastante los resultados iniciales, creando grupos no homogéneos y dispersos. Los resultados para calidad 85 % y calidad 80 % se pueden ver en las Tablas 6.5 y 6.6, respectivamente. Se observa claramente que mientras las imágenes originales podían ser agrupadas en sus respectivos grupos, al disminuir la calidad de la imagen y aumentar la compresión se pierde información relevante que conforma la huella PRNU y

dificulta la extracción de la misma.

Marca y Modelo	1	2	3	4	5	6	7	8
Canon 60D	24	2	4	1	2	1	1	0
Nikon D90	2	22	3	1	0	1	4	2
Nikon D7000	3	3	6	3	8	1	8	3
Sony A57	3	1	5	5	13	1	7	0

Tabla 6.5: Agrupamiento JPEG calidad 85 %

Marca y Modelo	1	2	3	4	5	6	7	8	9	10
Canon 60D	3	6	6	4	3	10	2	1	0	0
Nikon D90	12	4	2	0	4	2	1	5	0	5
Nikon D7000	1	1	0	3	6	13	2	3	5	1
Sony A57	3	0	0	5	13	4	1	6	2	1

Tabla 6.6: Agrupamiento JPEG calidad 80 %

6.2. Experimentos Agrupamiento de Vídeo

En esta sección se exponen los experimentos realizados para validar el algoritmo propuesto. Para ello, en el experimento 1 se demuestra experimentalmente que es necesario el uso de todos los fotogramas tipo I y no es suficiente el uso de *key-frames* puesto que se pierde mucha información. El experimento 2 compara el método de Calinski-Harabasz, el coeficiente silueta y el método del codo para la elección del número óptimo de grupos, concluyendo que el método del codo en este caso es el idóneo. Por último, en el experimento 3 se muestra un ejemplo completo del método utilizando los fotogramas tipo I y el método del codo, y se analiza la robustez del algoritmo mediante tres indicadores: la precisión, la exhaustividad y el valor-f.

Para los dos primeros experimentos se han utilizado 19 conjuntos de prueba generados con las combinaciones que aparecen en la Tabla 6.7, mientras que para el tercer experimento se ha generado un caso de prueba distinto.

Exp.	BQ Aquaris E5	Samsung G6	Iphone 7	Huawei Y635 L01	Iphone 8+	Nexus 5	Xiomi M3
1	5	5	5	0	4	0	0
2	8	8	0	8	7	0	0
3	0	0	0	0	5	5	0
4	7	7	7	0	7	0	0
5	5	5	5	0	5	0	0
6	4	4	4	0	3	0	0
7	9	9	9	9	8	0	0
8	6	6	0	6	5	0	0
9	6	6	0	6	6	6	6
10	5	5	5	5	5	5	0
11	5	5	0	5	0	5	0
12	9	9	0	9	0	9	0
13	9	9	0	9	0	9	0
14	0	5	5	5	0	5	5
15	0	5	5	0	0	5	0
16	6	6	6	0	5	6	0
17	0	0	0	5	5	6	0
18	0	5	6	4	0	3	6
19	6	5	0	0	0	3	0

Tabla 6.7: Datasets utilizados para identificación en vídeo

6.2.1. Experimento 1

El objetivo de este experimento es discernir si son necesarios todos los fotogramas tipo I para la obtención del PRNU del vídeo o por el contrario es necesario extraer solamente los *key-frames*, que contienen las diferentes escenas del vídeo y de esta forma no se produce un sesgo en la huella por las escenas. Si cierta parte del vídeo ha sido generada por escenas estáticas, el PRNU correspondiente a los fotogramas de dichas escenas predominará entre los ruidos extraídos por los fotogramas e impactará directamente sobre el del vídeo al promediarse estos.

Para poder comparar un método con el otro, es necesario aislarlo de la elección del número óptimo k de grupos, puesto lo que se busca es medir la calidad de las agrupaciones resultantes de los dos métodos. Por tanto, en ambos métodos se observará el agrupamiento con el número óptimo de clases ya que por una parte es imprescindible compararlos con el mismo k y por otra parte se deben comparar suponiendo el mejor caso de la elección del parámetro. Se utiliza la tasa de verdaderos positivos o TPR (del inglés *True Positive Rate*) como métrica.

Como se puede ver en la Tabla 6.8, el método que utiliza todos los fotogramas tipo I es más efectivo que el que utiliza los *key-frames*. De hecho, en ningún caso se obtiene un TPR mejor con los *key-frames*. Consecuentemente, se opta por la extracción del ruido de los fotogramas tipo I en el método propuesto.

6.2.2. Experimento 2

El objetivo de este segundo experimento es determinar el método a utilizar para la elección del número óptimo de grupos. Para ello se parte de la extracción del ruido de los fotogramas tipo I y se comparan el método de Calinski-Harabasz, el coeficiente silueta y el método del codo. No se ha incluido el método Gap en la comparación por su semejanza con el método del codo.

En la Tabla 6.9 se presenta para cada experimento el número verdadero de grupos y el número de grupos obtenido para cada uno de los métodos, resaltando en cada fila aquellos que han coincidido con el valor real. Como se puede ver, el método del

Experimento	N. Grupos	TPR Key-frames	TPR fotogramas tipo I
1	4	0.61	1
2	4	0.83	1
3	2	1	1
4	4	0.82	0.97
5	4	0.6	0.9
6	4	0.65	1
7	5	0.7	0.71
8	4	0.47	1
9	6	0.57	0.75
10	6	0.73	0.97
11	4	0.85	0.96
12	4	0.97	0.97
13	4	0.94	0.97
14	5	0.64	0.84
15	4	0.75	0.85
16	5	0.59	0.9
17	3	1	1
18	5	0.66	0.88
19	3	0.94	1

Tabla 6.8: Comparación extracción fotogramas tipo I vs. extracción *key-frames*

codo se comporta mejor que los otros dos métodos utilizados.

6.2.3. Experimento 3

En esta sección se analizan los resultados obtenidos a partir del algoritmo que ha mostrado ser más efectivo, mediante el uso de fotogramas tipo I y del método del codo para determinar el número de grupos óptimo. Para ello, se presentará la matriz de confusión y medidas como la precisión, la exhaustividad o el valor-f.

Para este experimento se ha utilizado un dataset compuesto por: 7 vídeos del dispositivo BQ Aquaris E5, 5 vídeos de un Iphone 7, 6 vídeos del Nexus 5, 4 del Samsung G6 y 5 del móvil Xiomi M3.

Tras realizar agrupamiento jerárquico se presenta en la Figura 6.1 la gráfica a utilizar en el método del codo, que representa el número de grupos frente a la distancia entre los dos grupos que han sido mezclados en ese paso. Se puede ver gráficamente como la distancia se reduce de forma drástica al aumentar el número

Exp.	N. Grupos	Calinski-Harabasz	Coeficiente silueta	Método del codo
1	4	2	**4**	**4**
2	4	2	2	**4**
3	2	**2**	**2**	**2**
4	4	2	**4**	**4**
5	4	2	6	3
6	4	2	2	**4**
7	5	2	3	6
8	4	2	2	**4**
9	6	2	2	4
10	6	2	3	**6**
11	4	2	2	3
12	4	2	2	**4**
13	4	2	2	**4**
14	5	2	3	**5**
15	4	2	3	**4**
16	5	2	7	**5**
17	3	2	2	**3**
18	5	2	2	4
19	3	2	2	**3**

Tabla 6.9: Comparación de métodos para la obtención del número óptimo de grupos

de grupos desde 1 a 3, y sigue reduciéndose de forma aproximadamente lineal hasta $k = 6$, donde k es el número de grupos. A partir de 6 grupos, la ganancia es mínima y se puede aproximar por una recta de pendiente cercana a 0. Por tanto, según el método del codo, el número óptimo de grupos es 6.

Por tanto, al cortar el dendrograma, ver Figura 6.2, de forma que queden 6 grupos se obtienen los resultados mostrados en la Tabla 6.10.

Dispositivo	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6
BQ Aquaris E5	7	0	0	0	0	0
Iphone 7	0	2	3	0	0	0
Nexus 5	0	0	0	6	0	0
Samsung G6	0	0	0	0	4	0
Xiomi M3	0	0	0	0	0	5

Tabla 6.10: Resultados agrupamiento

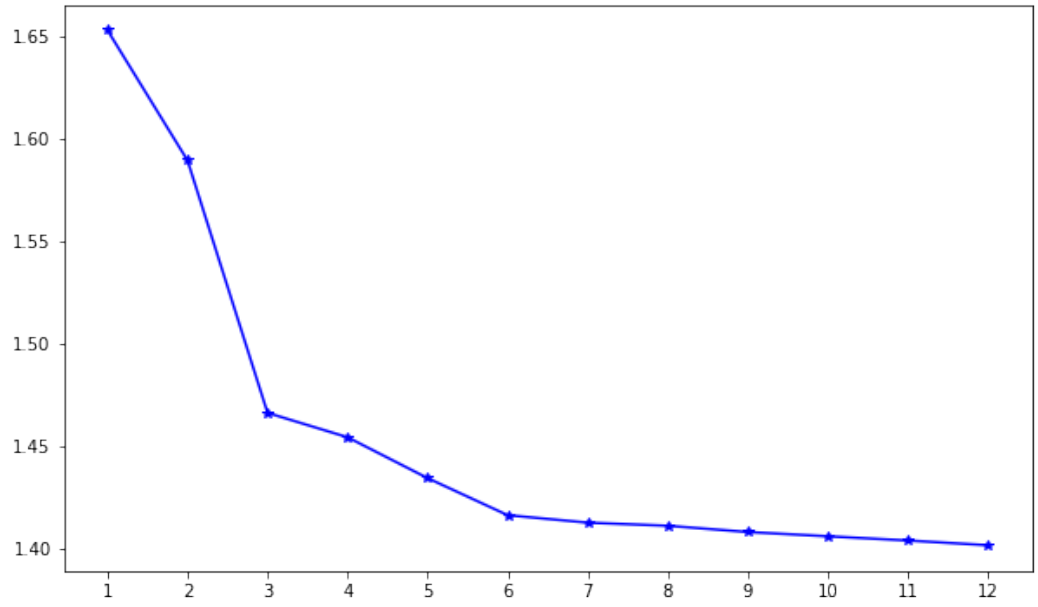


Figura 6.1: Número de grupos frente a la distancia de fusión

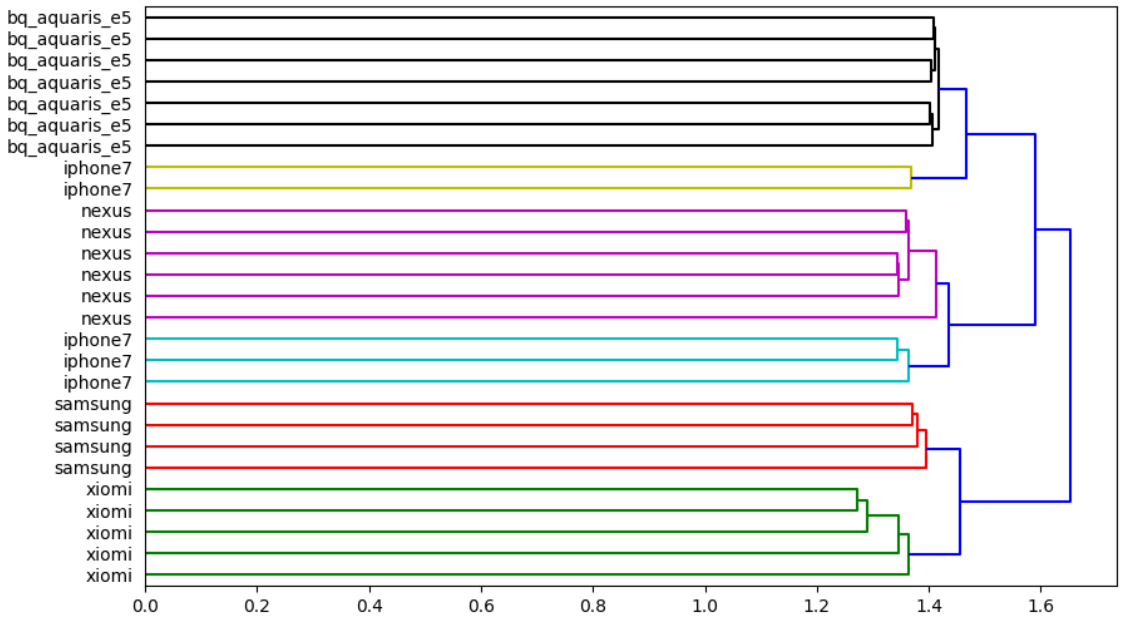
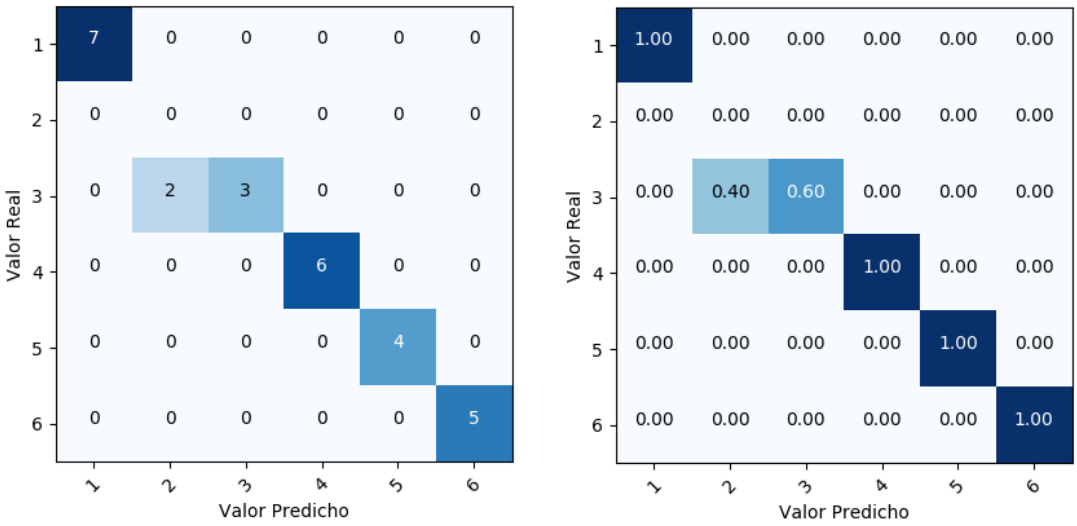


Figura 6.2: Dendrograma

Al haberse detectado a través del método del codo el número de grupos como 6, siendo 5 el real, los vídeos correspondientes al Iphone 7 aparecen dispersos en dos grupos distintos, mientras que el resto de dispositivos ha sido agrupado en grupos diferentes correctamente. Para construir las métricas que evalúan la calidad del agrupamiento, en primer lugar se debe construir la matriz de confusión. En este

caso se utiliza la matriz de confusión generalizada al trasladarse el problema de agrupamiento a un problema de clasificación multiclase.

A partir de la Figura 6.3 se calcula para cada grupo la precisión, la exhaustividad, el valor-f y el soporte, que es el número de elementos reales que pertenecen a un grupo.



(a) Matriz de confusión sin normalización (b) Matriz de confusión normalizada

Figura 6.3: Matrices de confusión

Promediando los valores de la Tabla 6.11 se obtiene que el agrupamiento resultante del algoritmo tiene una precisión de 1, una exhaustividad de 0.93 y un valor-f de 0.95.

Grupo	Precisión	Exhaustividad	Valor-f	Soporte
1	1	1	1	7
2	0	0	0	0
3	1	0.60	0.75	5
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1

Tabla 6.11: Métricas de evaluación del experimento

Si se compara este trabajo con el estado del arte se puede observar que la gran parte de trabajos no son válidos en este contexto ya que utilizan videos de mayor calidad procedentes de video cámaras. Más concretamente:

- En [44] obtienen precisiones del 100 % cuando utilizan videos procedentes de cámaras de video, pero esta precisión baja al 95 % cuando solamente utilizan videos de móviles.
- El algoritmo propuesto por [43] no es capaz de identificar como diferentes videos generados por dos móviles distintos del mismo modelo.
- La propuesta de [45] tiene la desventaja de tener una dependencia con la duración del video. En el caso de videos móviles la mayoría suelen durar apenas un par de minutos por lo que se hace poco efectivo en este contexto.
- En [46] obtienen una alta precisión pero el conjunto de prueba que utilizan procede de video cámaras, que generan video con mayor calidad por lo que no son equiparables.
- En [7] utilizan videos procedentes tanto de cámaras móviles como de vídeo cámaras.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

En este trabajo se ha propuesto una técnica de identificación de la fuente para vídeos en escenarios abiertos, en los que no se tiene ningún conocimiento de antemano ni del conjunto ni de otros dispositivos. El método se basa en características inherentes al proceso de fabricación de los sensores y las imperfecciones que este acarrea, por lo que es capaz de distinguir entre móviles del mismo modelo y marca. Para ello, extrae las características de los fotogramas del vídeo y selecciona los que contienen más información para promediar el ruido que se obtiene de cada uno mediante transformadas de ondícula. Por último, el algoritmo utiliza la huella extraída de cada vídeo como entrada en agrupamiento jerárquico y se elige el número óptimo de grupos mediante el método del codo.

Cabe destacar que a pesar de la numerosa literatura que hay en torno a la identificación de la fuente en imágenes, esta es escasa en el caso de vídeo. En este trabajo se presenta un algoritmo con una alta tasa de precisión y exhaustividad, muy similar o mejor que otras presentadas en trabajos relacionados. Sin embargo, el contexto del presente trabajo se ha restringido a vídeos generados por dispositivos móviles, que contienen menor información al estar más comprimidos que los generados por vídeo cámara, mientras que en el estado del arte la gran mayoría de ellos ha utilizado vídeos de este segundo tipo.

Según los experimentos realizados, el conjunto de frames de tipo I en su totalidad es necesario para obtener una huella apropiada de cada vídeo y no es suficiente con los key-frames. Además, el método del codo ha demostrado ser más eficaz que otros métodos ampliamente utilizados en agrupamiento como el índice Calinski-Harabasz o el coeficiente silueta, utilizado también en trabajos de identificación de la fuente en imágenes.

De esta forma, en este trabajo se presenta un método robusto para la identificación de la fuente en vídeo, compuesto de varias fases que han sido validadas experimentalmente de forma independiente, lo que da firmeza al proceso.

7.2. Trabajo Futuro

Se han identificado las siguientes líneas como posibles campos de ampliación o continuación de este trabajo:

- Comparar la transformada de ondícula de Daubechies con otras transformadas wavelet y analizar si se obtiene una huella mejor con otra descomposición.
- Analizar la calidad de la huella con I-frames en comparación a key-frames en vídeos con poco movimiento frente a otros con mayor movimiento. Ver si los key-frames se comportan mejor en el caso con menor movimiento y actualizar el algoritmo en base a esta información.
- Probar otro tipo de extracciones como el G-PRNU.
- Utilizar otro tipo de distancia en lugar de la correlación como puede ser la distancia de Mahalanobis o métodos de *block modeling*.

Capítulo 8

Introduction

Since the late 1950s, with the start of the Digital Revolution, numerous advances have taken place which have placed technology in a key and leading field. This has led to growing competitiveness in the sector that has resulted in better products, with lower costs, and constant innovations in the form of new components. In mid-2017, 5.7 billion unique mobile phone users were registered[1] with a projection that by 2020 three-quarters of the world's population would have at least one mobile phone.

Parallel to this progress, there has been a boom in the use of social networks and multimedia content. Only on YouTube, content equivalent to $46,000$ years, approximately one billion hours a day, is reproduced every year, 400 hours of new content is uploaded every minute and $70\,\%$ of the traffic is from mobile devices[2].

The wide use of mobile phones and the improvement and reduction of costs of photographic sensors have placed digital images and videos as one of the main and largest sources of data and information, which inevitably has meant an increase in image and video editing tools within the reach of everyone. In this way, numerous techniques of forgery and manipulation of multimedia content have arisen and with this a new field of forensic multimedia in continuous study and progress.

When videos are used as evidence in judicial processes, be it surveillance cameras or mobile devices, it is necessary to guarantee certain qualities of the same in order to be considered truthfully as evidence. One of these indispensable qualities is the

identification of the origin or authorship of the video or image, which could be comparable to ballistic weapons test.

As a result, forensic analysis has expanded its scope in recent years with the incorporation of images and video into daily life, with multimedia being a field of great relevance in various areas such as the judicial one.

In what follows, the object of the research is presented in 8.1. In 8.2 the context of the investigation is discussed and finally, in 8.3 the structure of the rest of the present work is described.

8.1. Objectives

Within the forensic multimedia analysis, numerous investigations have been carried out focused on the identification of the source in images, with excellent results. It could be considered that being a video formed by a set of images presented in a way that the brain perceives as continuous, there are also many other works focused on video with good results. However, the high levels of compression that exist when a video is generated entail a great loss of information that hinders the purpose of identifying the origin, so there is hardly any literature in this regard.

The identification of the source is based on unique imperfections that each sensor has, which directly affect the process of generating frames and can be extracted from them.

This work is focused on the extraction of strategic video frames and obtaining the fingerprint or noise of the sensor by means of wavelet transforms to identify the source in open scenarios, in which the set of devices is not known in advance nor does it have a database with footprints of certain devices.

8.2. Context

The present Final Master's Project is part of a research project entitled RAMSES approved by the European Commission within the Framework Program for Research and Innovation Horizon 2020 (Call H2020-FCT-2015, Innovation Action, Proposal Number: 700326) and in which the GASS Group of the Department of Software Engineering and Artificial Intelligence of the Faculty of Computing of the Complutense University of Madrid participates (Group of Analysis, Security and Systems, http://gass.ucm.es, group 910623 of the catalog of research groups recognized by the UCM).

In addition to the Complutense University of Madrid, the following entities participate:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (Spain)

8.3. Structure of the Work

The present work is organized in 7 chapters, this being the first one.

Chapter 2 introduces the essential concepts about videos, such as the process of creating a video based on sampling and quantification, storage by compression and extraction of noise in frames or images.

In Chapter 3, the forensic multimedia analysis and its different techniques are presented. Specifically, the state of the art is detailed involving the extraction of key-frames, the detection of manipulations or double compression and source identification.

Chapter 4 deals with clustering algorithms. It introduces two different types of clustering, K-means and hierarchical clustering, different methods for choosing the optimal number of clusters and evaluation metrics in clustering based on multiclass classification.

Chapter 5 describes the contribution made, that is, an algorithm for extracting frames with a high degree of information and a clustering algorithm based on the extraction of sensor noise by means of the Daubechies wavelet transform that allows the identification of the video source.

The experimentation that validates the proposed method is detailed in Chapter 6.

Chapter 7 indicates the conclusions drawn from this work and the future lines of research in this field.

Capítulo 9

Conclusions and Future Work

9.1. Conclusions

In this work a technique for identifying the source for videos in open scenarios has been proposed, in which there is no knowledge in advance of the set or other devices. The method is based on inherent characteristics to the manufacturing process of the sensors and the imperfections that this entails, consequently it is capable of distinguishing between mobile phones of the same model and brand. To accomplish this, it extracts the main characteristics of the frames of the video and selects those that contain more information, to average the noise that is obtained from each one through wavelet transforms. Finally, the algorithm uses the fingerprint extracted from each video as input in hierarchical clustering and the optimal number of clusters is chosen using the elbow method.

It should be noted that despite the numerous literature that exists involving the identification of the source in images, it is scarce in the case of video. In this paper an algorithm with a high rate of precision and recall, very similar or better than others presented in related works, is presented. However, the context of this work has been restricted to videos generated by mobile devices, which contain less information as have higher compression than those generated by video cameras, while in the state of the art the vast majority of studies have used videos of this second type.

According to the experiments carried out, the whole set of I-frames is necessary to obtain an appropriate footprint of each video and it is not enough with only the key-frames. In addition, the elbow method has proven to be more effective than other methods widely used in clustering, such as the Calinski-Harabasz index or the silhouette coefficient, which is also used in other studies to identify the source in images.

In this way, this paper presents a robust method for the identification of the video source, composed of several phases that have been validated experimentally and independently, which adds firmness to the process.

9.2. Future Work

The following lines of work have been identified as possible fields of extension or continuation of this one:

- Compare the Daubechies wavelet transform against other wavelet transforms and analyze if a better sensor noise is obtained with another decomposition.
- Analyze the quality of the footprint using I-frames compared to key-frames in still videos and non-still videos. See if the key-frames behave better in still videos and update the algorithm based on this information.
- Try other type of extractions such as G-PRNU.
- Use another type of distance instead of the correlation such as Mahalanobis distance or methods of *block modeling*.

Bibliografía

[1] GSMA (2017). The Mobile Economy 2017. Available: https://www.gsma.com/newsroom/press-release/number-of-global-mobile-subscribers-to-surpass-five-billion-this-year/

[2] DMR Business Statistics (2018). 160+ YouTube Stats and Facts (August 2018) By the Numbers. Available: https://expandedramblings.com/index.php/youtube-statistics/

[3] J. Lukas, J. Fridrich, M. Goljan (2006)."Digital camera identification from sensor pattern noise". *IEEE Transactions on Information Forensics and Security, 1(2), 205-214.*

[4] Daubechies wavelet. Available: https://en.wikipedia.org/wiki/Daubechies_wavelet

[5] K. Dabov, A. Foi, V. Katkovnik *et al* (2007). "Image denoising by sparse 3-d transform domain collaborative filtering". *IEEE Trans. Image Process., 16(8), pp. 2080-2095*

[6] F. Gisolf, A. Malgoezar, T. Baar, *et al* (2013). "Improving source camera identification using a simplified total variation based noise removal algorithm". *Digital Invest., 10(3), pp. 207-214*

[7] M. Al-Athamneh, F. Kurugollu, D. Crookes, M. Farid (2016). "Digital video source identification based on green-channel photo response non-uniformity (G-PRNU)". *Sixth International Conference on Computer Science, Engineering & Applications*

[8] T. Filler, J. Fridrich, M. Goljan (2008). "Using sensor pattern noise for camera model identification". *15th International Conference on Image Processing, San Diego, CA, 2008, pp. 1296-1299*

[9] A. Murat (2015). "Digital Video Processing, Second Edition". *Prentice Hall Signal Processing*

[10] A. Bovik (2005). "Handbook of Image and Video Processing, Second Edition". *Academic Press*

[11] M. Parker, S. Dhanani (2012). "Digital Video Processing for Engineers". *Newnes*

[12] Wiener Filter. Available: https://en.wikipedia.org/wiki/Wiener_filter

[13] K. Sowmya, H. Chennamma (2015). "A survey on video forgery detection". *International Journal of Computer Engineering and Applications, Volume IX*

[14] P. Bestagini, M. Fontani, S. Milani, M. Barni *et al* (2012). An overview on video forensics. *APSIPA Transactions on Signals and Information Processing, V1, pp. 1229-1233*

[15] N. Mondaini, R. Caldelli, A. Piva, M. Barni *et al* (2007). Detection of malevolant changes in digital video for forensic applications. *Proc. of SPIE, Security, Steganoprahy, and Watermarking of Multimedia Contents IX, E.J.D III and P. W. Wong, eds., vol. 6505, no. 1, SPIE, 65050T*

[16] M. Kobayashi, T. Okabe, Y. Sato (2010). Detecting forgery from static-scene video based on inconsistency in noise levels functions. *IEEE Trans. Info. Forensic Secur., 5(4). pp. 883-892*

[17] V. Conotter, J. O'Brien, H. Farid (2011). Exposing digital forgeries in ballistic motion. *IEEE Trans. Info. Forensics Secur., pp 99*

[18] J. Fridrich (1998). *Image watermarking for tamper detection. ICIP (2), pp. 404-408*

[19] J. Lukás, J. Fridrich (2003). Estimation of primary quantization matrix in double compressed jpeg images. *Proc. of DFRWS*

[20] S. Milani, M. Tagliasacchi, M. Tubaro (2012). Discriminating multiple jpeg compression using first digit features. *Proc. of the 37th Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 2253-2256*

[21] W. Wang, H. Farid (2009). Exposing digital forgeries in video by detecting double quantization. *Proc. 11th ACM Workshop on Multimedia and Security, MM&Sec '09, ACM, New York, NY, pp. 39-48*

[22] W. Wang, H. Farid (2009). Exposing Digital Forgeries in Video by Detecting Double MPEG Compression. *MM&Sec'06 Proc. of the 8th workshop on Multimedia and Security, pp. 37-47*

[23] K. Sayood (2012). Introduction to Data Compression, 4th Edition. *Morgan Kaufmann*

[24] L.J. García Villalba, A. Lucila Sandoval, J. Rosales Corripio (2015). Smartphone image clustering. *Expert Systems with Applications, 42, pp. 1927-1940*

[25] Codificación Huffman. Available: https://en.wikipedia.org/wiki/Huffman_coding

[26] G. Lin, J. Chang (2013). Detection of Frame Duplication Forgery in Videos based on Spatial and Temporal Analysis. *International Journal of Pattern Recognition and Artificial Intelligence*

[27] Z. Huang, F. Huang, J. Huang. Detection of double compression with the same bit rate in MPEG-2 videos. *IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*

[28] H. Yao, S. Song, C. Qin, Z. Tang, X. Liu (2017). Detection of Double-Compressed H.264/AVC Video Incorporating the Features of String of Data Bits and Skip Macroblocks. *Symmetry*

[29] D. Vázquez-Padín, M. Fontani, T. Bianchi, P. Comesaña *et al* (2012). Detection of video double encoding with GOP size estimation. *IEEE International Workshop on Information Forensics and Security (WIFS)*

[30] W. Wang, X. Jiang, S. Wang, T. Sun (2013). Estimation of the primary quantization parameter in MPEG videos. *Visual Communications and Image Processing (VCIP)*

[31] Y. Wu, X. Jiang, T. Sun, W. Wang (2014). Exposing video inter-frame forgery based on velocity field consistency. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pp. 2674-2678*

[32] Grubb's test. Available: https://en.wikipedia.org/wiki/Grubbs'_test_for_outliers

[33] Q. Wang, Z. Li, Z. Zhang, Q. Ma (2014). Video Inter-Frame Forgery Identification Based on Consistency of Correlation Coefficients of Gray Values. *Journal of Computer and Communications, 2, pp. 51-57*

[34] Q. Wang, Z. Li, Z. Zhang, Q. Ma (2014). Video Inter-Frame Forgery Identification Based on Optical Flow Consistency. *Sensors & Transducers, 166, pp. 229-234*

[35] D. Fu, Y. Q. Shi, W. Su (2009). A generalized benfords law for jpeg coefficients and its applications in image forensics. *Proc. of SPIE, Security, Steganography and Watermarking of Multimedia Contents IX, vol. 6505, pp. 39-48*

[36] R. C. Reininger, J. D. Gibson (1983). Distributions of the two dimensional DCT coefficients for images. *IEEE Trans. On Commun., vol. COM-31, pp. 835-839*

[37] J. D. Eggerton, M. D. Srinath (1986). Statistical distribution of image DCT coefficients. *Computer and Electrical Engineering, vol. 12, pp. 137-145*

[38] D. B. Tariang, A. Roy, R. S. Chakraborty, R. Naskar (2017). Automated JPEG forgery detection with correlation based location. *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*

[39] X. Jiang, W. Wang, T. Sun, Y. Q. Shi *et al* (2013). Detection of Double Compression in MPEG-4 Videos Based on Markov Statistics. *IEEE Signal Processing Letters*

[40] W. Chen, Y. Q. Shi (2009). Detection of double MPEG compression based on first digit statistics. *Lect. Notes Comput. Sci. (IWDW 2008), vol. 5450, pp. 16-30*

[41] S. Jia, Z. Xu, H. Wang, C. Feng, T. Wang (2018). Coarse-to-fine Copy-move Forgery Detection for Video Forensics. *IEEE Access*

[42] S. Kingra, N. Aggarwal, R. D. Singh (2017). Video Inter-frame Forgery Detection Approach for Surveillance and Mobile Recorded Videos. *International Journal of Electrical and Computer Engineering (IJECE), vol. 7, no. 2, pp. 831-841*

[43] Y. Su, J. Xu, B. Dong, J. Zhang (2010). A novel source MPEG-2 video identification algorithm. *International Journal of Pattern Recognition and Artificial Intelligence, vol. 24, no. 8, pp. 1311-1328*

[44] S. Naveen, J. A. Reyaz, C. Balan (2016). Video Source Identification. *International Journal of Computer Science and Information Technologies (IJCSIT), vol. 7 (1), pp. 363-366*

[45] M. Chen, J. Fridrich, M. Goljan, J. Lukás (2007). Source Digital Camcorder Identification Using Sensor Photo Response Non-Uniformity. *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, 65051G*

[46] S. Yahaya, A. TS Ho, A. A. Wahab (2012). Advanced video camera identification using conditional probability features. *Proc. of the IET Conference on Image Processing, pp. 1-5*

[47] Y. Su, J. Xu, B. Dong (2009). A source video identification algorithm based on motion vectors. *Proc. of the Second International Workshop on Computer Science and Engineering, vol. 2, pp. 312-316*

[48] R. Tibshirani, G. Walther, T. Hastie (2001). Estimating the number of clusters in a dataset via de Gap statistic. *Journal of the Royal Statistical Society Series B (Statistical Methodology), vol. 63 (2), pp. 411-423*

[49] A. L. Sandoval, L. J. García Villalba, D. M. Arenas, J. Rosales *et al* (2015). Smartphone Image acquisition Forensics using Sensor Fingerprint. *IET Computer Vision, vol. 9 (5), pp. 723-831*

[50] A. K. Jain, A. Vailaya (1996). Image retrieval using color and shape. *Pattern recognition, vol. 29 (8), pp. 1233-1244*

[51] S. Jeong (2001). Histogram-Based Color Image Retrieval. *Psych221/EE362 Project Report, Stanford University*

[52] M. Zhang, L. Tian, C. Li (2017). Key frame extraction based on entropy difference and perceptual hash. *IEEE International Symposium on Multimedia*

[53] J. Segers (2014). Perceptual image hashes. Available: https://jenssegers.com/61/perceptual-image-hashes

[54] J. B. MacQueen (1967). Some methods for classification analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*

[55] D. Arthur, S. Vassilvitskii (2007). k-means++: the advantages of careful seeding. *SODA '07 Proceedings of the eighteenth annual AMC-SIAM symposium on Discrete algorithms, pp. 1027-1035*

[56] G. Hamerly, C. Elkan (2002). Alternatives to the k-means algorithm that find better clusterings. *CIKM '02 Proceedings of the eleventh international conference on Information and knowledge management, pp. 600-607*

[57] Rokach, Lior, O. Maimon (2005). Clustering methods. *Data mining and knowledge discovery handbook, Springer US, pp. 321-352*

[58] Scipy. Distance computations. Available: https://docs.scipy.org/doc/scipy/reference/spatial.distance.html